Command & Control:
Understanding, denying, detecting
People Who Know How

To Protect What’s Important To You

QinetiQ Proprietary
QINETIQ/TIS/SEC/TR1303462
© QinetiQ Ltd 2014
1 Summary

Modern computer usage has seen an ever-increasing use of the Internet. More and more business is being conducted on or over the Internet, communication via the Internet is the norm for many transactions, and many people now socialise via sites on the Internet. This is generally seen as a beneficial progression of technology; however, any progress can be used negatively as well as positively, and so the Internet is also used as a conduit for malware and crime.

Many computer users are familiar with stories of indiscriminate malware whose purpose is to steal credentials, blackmail users or just cause disruption on a network. This class of malware can be described as a “fire-and-forget” attack, where no further instruction is required from the malware author after initial release. BotNets are familiar to many users as a network of infected machines that are controlled for the purposes of sending spam and other attacks. More complex and long-lived malware with more specific goals has recently emerged as the major threat to many organisations, often referred to as Advanced Persistent Threats (APTs). BotNets, APTs and other prolonged attacks require further instruction and remote control to be successful, including information such as where to attack or which machines to collect information from. This prolonged control requires some form of communication channel over which commands and results can be sent. This is known as a Command and Control (C2) channel. In addition to this channel, APT attacks will often have a data exfiltration channel that may or may not use the same mechanism as the C2 channel.

The main body of this document briefly describes the threats to the networked world that we live in, concentrating particularly on the C2 channel of malware and APTs. We describe what C2 achieves, how C2 works, how C2 can be defended against, and examples of C2 channels that we have seen in the wild. The annexes provide additional detail on these areas: Annex A (Command & Control Channels) provides additional technical detail on the protocols used by C2 channels and how they are developing; Annex B (C2 and the Council on CyberSecurity’s Top 20 Controls) provides a review of the Council on CyberSecurity’s Top 20 Critical Security Controls that relate to C2 channels and suggests some additional controls.

We conclude that the direct detection of APT C2 channels is difficult and is always subject to change and innovation by the attacker. Detection is best achieved by examining communication patterns over many nodes over an extended period rather than micro-examination of specific packets or protocol patterns for malware. This is because alerting on specific protocol patterns tends to generate too many “false positives” to be useful. Nevertheless, certain aspects of APT behaviour, especially network reconnaissance and bulk data exfiltration – can be detected by observing trends over periods of days or weeks to spot unusual patterns. From this conclusion we present advice on the best approaches for detecting and blocking such advanced attack, in addition to the Council on CyberSecurity’s Top 20 Critical Security Controls.

QinetiQ would like to acknowledge the help and support of CPNI in producing this Command and Control document and the accompanying material.
Organisations spend significant sums of money each year on security appliances and security software. Security appliances - including email and web filtering systems, firewalls and software security products such as on host Anti-Virus (AV) and Host Intrusion Prevention Systems (HIPS) - will absorb most of an organisation’s security budgets. Security architects build these components into corporate networks to provide layers of defences in a bid to prevent malicious software from entering their networks, and to detect any outbreak of infection.

In an ideal world these appliances and software products would prevent all attacks. However, in reality the best AV products have a successful detection rate of around 99% [1] for known malicious software. This detection rate falls sharply when examining new malware [2] and falls further when examining bespoke, targeted malware and when false-positives are removed [3]. Similarly, other security appliances are only as effective as their rule configurations.

We refer to the current state of affairs as a “post-exploitation” world, where infection is the norm rather than the exception. It makes little difference if operating systems and third party applications are patched and up-to-date, as there are no security appliances or technical controls which can stop a user community from being coerced into bypassing security procedures. Using techniques like social engineering, USB key drops, and spear phishing, it is highly likely an attacker will be able to circumvent an organisation’s security boundaries and execute their payload upon the systems. Taking advantage of bad user behaviours was a contributing factor in the spread and success of the Stuxnet worm, for example.

To further compound this problem, some organisations have begun to adopt the Bring Your Own Device (BYOD) model where users can connect their personal mobile phone, laptop, tablet, or other devices to a corporate network. Indeed, another attack surface where hosts may be compromised is when users Take a Business Laptop Home (TBLH) or connect their corporate device to a hotel, home, or another unknown wireless or wired network. In the BYOD world, we have to assume the device has been compromised and there is potentially a hostile presence. Similarly, with TBLH, we can assume that any foreign network environment that your controlled device is exposed to is also hostile.

We live in a world where corporate systems do get compromised, often through unexpected or unanticipated attack paths. This complex, novel and unexpected type of malware infection is often termed as an Advanced Persistent Threat (APT) due to the method of delivery, exploitation and unknown malicious intent of the payload. The goal cannot realistically be to prevent 100% of all infections, but instead it should be to implement all reasonable measures (such as those embodied in the Council on CyberSecurity’s Top 20 Critical Security Controls – detailed further in Annex B) to limit the infections but then to accept the inevitable compromise and to be prepared to deal with it using appropriate disinfection and recovery procedures.
The modern cyber security domain has seen increasing concern about complex and well-funded APTs that “play the long game” of compromising a network and slowly undertaking their mission. This increased concern about advanced attack has been mirrored by an increasing use of military terminology to describe aspects of the attacks.

The lifecycle of malware, particularly APTs and espionage-driven attacks, can be described in the form of a “kill chain”. A kill chain is a phrase taken from military contexts, where it describes the process of identifying, understanding, and attacking a target of interest. In the cyber security domain the phrase has been used to describe the process of identifying a target network, understanding the network, attacking and then exploiting the network.

Command and Control (shortened to C2) is another military term that has become part of the vocabulary of the cyber security domain. C2 is the direction of forces by a commanding officer to accomplish a mission, and is applied to the direction of malware by its controller1.

Many security-related organisations have their own minor variations on the kill chain. Lockheed Martin have produced one kill chain that focuses on the early steps of the process [4]. This is just one of many ways to break down the kill-chain, and focuses on the early steps of exploitation, prior to actual infection. QinetiQ use a slightly different kill chain that focuses more on the later exploitation phases of the attack, as seen in Figure 2, below. Although there are differences, the two kill chains can be mapped to each other.

---

1. It is important to note that in the modern world, where malware is readily available for sale, then the author of a malware program may not necessarily be its operator.
Within both of the kill chains, the first step is a process of reconnaissance, using openly available information, social engineering and subtle probes to understand the target and its defences. Having understood the defences, the attacker then proceeds to “weaponise” their knowledge to deliver an attack that should bypass the known defences without triggering any alerts. Having delivered the attack, it will then be executed, leading to exploitation of the system and installation of a “beach-head” within the network. Once the system has been exploited and the malware is installed then the mission of the attack is undertaken in the remaining phases under the control of the attacker, using C2 channels to send commands and retrieve data. Here, the emphasis of the QinetiQ kill-chain on the later phases of the attack separates the attacker’s C2 control step from the activity of the malware on the infected computer actioning those commands before egressing data and attempting to persist on the network and awaiting future commands.

For a number of malware attacks, such as the capture of banking or other valuable credentials (such as computer game accounts) then the kill-chain is still relevant, but no C2 will happen and the attack will effectively be a one-time attack. The malware infection may remain and continue to capture credentials before it egresses the data to a pre-defined drop-off point, but the activity will not be directed beyond the initial programming and it will not change its behaviour. By contrast, botnets, Trojans and APTs rely on a communication channel to provide new instructions for a compromise to be completed successfully, thereby making them continually able to change their behaviour (within the constraints of their programming).

If a network defender can stop the attack at any point in the main kill-chain then the attack will be rendered ineffective (although at any point after a successful delivery then the attack could subsequently be resurrected and the threat has not completely subsided). However, it is also true that the defender needs to have a perfect defence record to remain uncompromised, whereas the attacker only needs one successful compromise to have achieved their aims.

Directing malware in this way once an infection has been achieved requires a C2 channel, but merely retrieving data from pre-defined locations or processes and transferring it to a location on the Internet that is accessible to the malware controller does not necessitate a C2 channel. Credential-stealing malware needs to return data but not receive new commands and so does not require a C2 channel. APTs must be commanded to perform further actions based on the information that they discover and so do establish C2 channels. Where data is retrieved by a C2-controlled infection, the channel used may be the same as the command channel but a separate channel could be used.

Having taken the terminology from the military domain to describe what the attacker needs to do (controlling their malware infection), the following section describes how these C2 channels can be established in reality.
4 C2 Methods

At its most abstract level, C2 methods can be split into two categories: Push and Pull. These two categories only differ in how new commands are delivered to the compromised computers, but not in the commands that can be delivered. In addition to the command system, an attacker will need to receive feedback and (optionally, depending on the goal) make unauthorised transfers of valuable intellectual property and data from the compromised network in a process known as exfiltration. This feedback could be via the same mechanism as the commands or a separate channel may be used, particularly where the command mechanism is suitable for small messages but the feedback is in the form of exfiltrating large files.

Regardless of the channel used, malware with a C2 channel tends to follow a similar traffic pattern: registration of a new infection or resumption of an existing infection session; transfer of commands to be executed; exfiltration of results from any commands executed. The last two steps are then repeated until either the compromise is discovered and removed, or the attacker’s goal has been satisfied and the compromise is either deleted from the system or left as a future back-door.

Push model C2 channels work by having the malware connect out to a known host so that commands can be sent to it in near real-time, as if the attacker was operating the compromised computer. This is analogous to most voice communication between humans.

Pull model C2 channels work by using known drop-points that are regularly checked for new commands by the malware. The locations of the drop points, or the algorithm for generating them, are determined in advance. If no new commands are available then nothing will be found at the drop point and malware will continue with its existing commands until the next check-in time. This technique has commonly been used during the Cold War to allow spies and informants to communicate without meeting face-to-face.

![Figure 3 - Traffic pattern of C2 compromise](image)

**Push Model**
1. Attacker sends commands and receives results
2. Compromised Computer
3. C2 Control Hub

**Pull Model**
1. Attacker leaves commands at hub
2. Machine periodically retrieves commands
3. Machine leaves results on hub
4. Attacker periodically retrieves results

![Figure 4 - Communication pattern of Push and Pull Model C2 Channels](image)

Just as there are many ways in which humans can communicate², so there are many ways in which a malware controller can communicate with the compromised computers. For each way in which humans communicate then there is a comparable protocol that works in a similar fashion. Just as any human conversation can be benign or malicious, so these machine to machine protocols can be used for malicious purposes.

2. Initial thoughts may be that this list is short, but a non-exhaustive enumeration includes: Face-to-face vocal discussion, sign-language, telephone, video, radio, Morse code, written letters, telegram, chalked marks and other inscriptions in public locations, and more recently through RFID or 2D barcodes.
Modern malware has been identified using a variety of protocols to implement a C2 channel and has been seen to use a variety of techniques to disguise itself. More detail can be found in Annex 1.

As there are methods that are currently the primary means of running C2 channels, it can be assumed that their rate of success is sufficiently high on the networks that have been targeted for the malware authors to continue using them. However, should these methods become sufficiently compromised – whether by tighter security or faster response by the owners of the infected networks – then new methods will be developed. Some methods for C2 have already been explored by researchers but are not known to have been seen in the wild. Furthermore, there are undoubtedly more currently unforeseen methods and protocols that could be used as C2 channels. More detail on current and potential future C2 methods is provided in Annex A.

Given this variety of methods of establishing a C2 channel, some of which use commonly used protocols and media (such as postings on social media sites), it is not possible to build a network that is both useful for business and completely secured against these attacks. However, the next section describes what can be done to defend a network.
5 Detection, Disruption and Defences

As described in the previous section, the detection of some C2 channels is very difficult due to the nature of the channel and the accurate blocking of all such channels is even more so. While this does make the defence of a network against C2-controlled malware more difficult, it does not make it impossible or worthless. The rest of this section looks at how C2 channels might be disrupted and how networks can best be defended.

As an example of the difficulty of detecting some C2 channels, the process of looking for malicious social media messages amongst the mass of other social media messages is akin to looking for a small, yellow needle in a very large haystack. While it may be possible to create a procedure or tool to detect such a needle, the methods used by the attacker are such that they will change the shape and colour of the needle for future attacks, thereby making the defender’s new technique ineffective.

For the most part, malware that involves the use of C2 channels is similar to non-C2 malware at the initial point of infection and so all of the conventional defences are applicable to protection against it. There is no “silver bullet” of defending against malware or C2 channels, but anti-virus, Intrusion Detection Systems (IDSe), Internet and email traffic filtering and scanning, blacklistng of known malware hosts, appropriate security training of all staff to reduce the success of phishing attacks and other measures can all limit the likelihood of initial compromise. No defences are foolproof, though, and it is important to remember that the evolution of malware means that those tools that provide the best defence today may not provide as much protection in the future.

5.1 Disruption by Technical Defence

Despite the difficulty of defeating C2, there are some ways in which some current C2 channels can be disrupted or prevented. At its simplest level, implementing network security best practices, as captured by the Council on CyberSecurity Critical Security Controls list (see Annex 3 for more detail) will defend a network against a subset of malware and some, but not all, C2 channels. A network that is configured with proxy servers and firewall rules to prevent communication on all ports that are not required for normal business operation can prevent unexpected direct communication from internal machines to external machines. This would block simple attempts to connect to chat servers from a network that has no need for chat protocols, or direct HTTP connections that do not use the appropriate proxy server, and would provide logging of activity that did use the proxy servers.

For core protocols that cannot generally be configured to use a proxy server, a more advanced approach is to lock down certain services to certain machines – a firewall that only allows one known DNS server to connect externally to upstream DNS servers will prevent direct DNS lookup attempts, and any queries that follow the correct channel out of the network can more readily be logged and examined.

An extension to the use of firewalls for filtering is the use of protocol anomaly detection. This is required because the allocation of ports is arbitrary, and hence any service can be run over any port, but certain ports are “reserved” for use by specific protocols due to existing conventions. This technique aims to protect against methods such as running IRC over HTTP’s port 80 by understanding the protocols that should be crossing a perimeter and blocking any content that does not meet the required structure of the expected protocol. Similarly, network communication may be broken down at the packet level and reassembled to create protocol break, resulting in packets that are known to comply with the protocol requirements. Where the risk is high enough and the protocol break appliance is suitably assured then this could provide valuable additional security.

Although such filtering will detect and disrupt C2 channels that use unconventional protocols running over standard ports in an attempt to hide their activity, it will not detect when C2 channels use protocols that are allowed and required for business communication. There is nothing inherently unusual, for example, about unencrypted HTTP traffic over port 80 or encrypted HTTP traffic over port 443, but content analysis may be able to detect these channels. Where C2 channels also try to obfuscate their messages (either through encryption or hiding content within seemingly legitimate content) then content checking may not be sufficient. In these cases, reformatting, restructuring, normalising and re-encoding the data may disrupt or block the channel. The potential issue with this defence is that as the communication medium is also used for corporate use then any false-positives will have a negative impact on the business due to breakages and downgrading in the quality of data received. Further, any re-encoding of content will incur additional latency and processing overhead.
All of these techniques can be used to entrench best practice in network design and prevent some C2 channels, as shown in Figure 5, above (also including monitoring). Here there are a number of defensive measures in place on a network, highlighted in blue. Each desktop on the monitored network contains a remote forensic agent that monitors the node’s behaviour. The network also contains honeypots and honeynets to catch attacks that are exploring the network. The protocol proxies and ingress/egress protection allow filtering and detection of external traffic that may contain C2 channels. Finally, the network taps and APT Investigation Workbenches, together with the Analysis Centre monitor the network to detect malware and C2 channels that are either attempted and disrupted or successfully created, which is discussed in the following section.

Due to the variety of techniques that could be employed, and the possibility that the traffic is indistinguishable from legitimate traffic at the level of an individual packet, it is not possible to block all C2 channels in this way. In these cases then an alternative, indirect technique of tackling the threat of C2 channels must be employed.
5.2 Defence by Monitoring

Monitoring is a tool to defend against on-going C2 channels, and malware infection in general, that complements the technical defences of the previous section. In those cases where the C2 channel cannot be directly detected, the impact of the action may be observable in the same way that ripples on a pond can be seen, even if the object that the ripples went unnoticed.

Short-term monitoring for deviations from the norm – which examines data covering periods from days down to minutes - is inherently likely to raise false-positives because of the variance in behaviour of users and machines across different times of the day and different days of the week. If a system is instead monitored for an extended period of time – ideally for multiple months – then deviations from the norm can become more prominent while background noise is averaged out, thereby aiding in the identification of infected hosts.

Even in situations where disruptive defences are in place, a network should still be monitored against attempts to create C2 channels. When a network disrupts or blocks the C2 channel of a successful malware infection then the malware cannot connect to its controller. This will prevent it receiving direction for new tasks or returning information that would influence subsequent commands, but the network itself will still be infected and the malware still poses a risk. Just as quarantine does not end the risk of infection, so a malware infection that has compromised hosts but not established a C2 channel is still a danger. If the infected hosts were configured with a set of preliminary, undirected tasks then they will still be active and will be waiting to take advantage of any short chink in the network’s defences that may be caused by a misconfiguration, an upgrade or another unexpected avenue. Without appropriate monitoring, the compromise would not be seen and clean-up could not be directed.

There are many tools and techniques for monitoring networks and detecting attacks that can be useful against APTs and the kinds of malware that use C2 channels. Honeypots and honeynets provide fake systems that legitimate systems will never connect to, but which malware may discover and attempt to exploit. Honey-tokens are text strings and data that should never leave a network but which perimeter measures have been configured to detect leaving the network, such as when malware exfiltrates what its controller believes is valuable data. Some Intrusion Detection System (IDS) signature sets include rules for “emerging threats” that attempt to identify behaviour that would be undesirable on a corporate network and that might be indicative of C2 channels and malware exploitation, including checking communication against lists of IP addresses that are known to host malware or C2 control nodes. Finally, the GovCert IDS rules are similarly useful for organisations that have access to them.

5.2.1 Full Data-style Analytics for Monitoring

All of the data collected from conventional logging methods and from these additional tools can add up to vast data sets. To best analyse these data sets for signs of malware compromise (either unseen traditional fire-and-forget malware or more complex APT-style malware that may be using C2 channels) over extended periods then alternative methods of data storage and processing may be required. If tools do not use these methods then the analysts may be overwhelmed and important signs of malware infection and C2 channels may be missed.

For a network that is generating hundreds of gigabytes or more of data each day, a longer-term view of network traffic patterns may necessitate “big data” storage and analysis techniques, but smaller networks may be able to achieve the same effect using more conventional technologies. The advantage of big data tools are that they can help to handle the large volumes and semi-structured data formats involved in monitoring large networks without constraining the advanced analysis to only those methods that have been configured in advance.

As part of this study, QinetiQ used a set of techniques based on these full data processing ideas and influenced by Big Data thinking. The techniques can be summarised as follows:

- Correlations: using multiple sources in combination to detect suspected attack
- A skilled human analyst to drive the tool and interpret the results
- Data Analytics: analytic functions that operate over potentially large datasets to identify patterns, highlighting things that are anomalous. A Big Data repository or conventional database can be used to support the analytics.

These techniques do not do away with the need for conventional defences, such as firewalls and antivirus, nor do they obviate the need for traditional protective
monitoring and SIEM (Security Incident and Event Monitoring) tools. The need for such defences is greater than ever. The techniques described here are complementary approaches that must work in conjunction with conventional defences. A note of caution is required, though, in that each organisation implementing such “full data capture” monitoring must individually ensure that it is appropriately compliant with its network procedures and local law (particularly data protection law, which is especially relevant if staff are allowed to use the network for occasional personal use), and that staff are aware of the monitoring activity and its implications on their communication across the network.

The human eye is very good at spotting patterns, and as such a monitoring system that captures, parses and presents the data to an intelligent human analyst (such as the example shown in Figure 6, below) can help to identify trends and patterns that indicate a compromise that could be an APT being controlled via a C2 channel, even where the channel itself has not been spotted or disrupted. It is better to detect the compromise and be unaware of the C2 channel than to assume the network is clean because no C2 communication has been seen.

While this monitoring can be built on top of simple metrics, our finding is that the greatest value comes not from comparing and placing thresholds on absolutes but instead by comparing relatives and using holistic measurements. Also, it is important to understand the network that is being monitored and what “normal” might look like on that network so that known “noise” can be filtered out. Without this underlying knowledge of what is expected on the network then it becomes more difficult to recognise potential malware behaviour amongst the commonly observed network behaviour.

A prime example of where network knowledge is required is in the analysis of user agents provided with requests (the strings that browsers and other tools use to tell a web server what application they are). There are many normal user agents that would be expected on a network (such as Internet Explorer, Mozilla Firefox and others) and any non-standard user agents should be considered suspicious. While malware authors could identify their programs as conventional web browsers, they frequently do not. As the user agent is an arbitrary string that is provided for reference then it is also a potential exfiltration channel, as data could be encoded and stored within it. In an trial sample of user agents seen on a network, a number stood out as appearing to be encoding data, but further investigation showed that the data was being transmitted to known Anti-Virus vendors as part of their expanded protection suite. These user agents provided a lot of noise in the data, which had to be filtered out while carefully retaining other anomalous user agents. This data could only be properly understood and analysed by knowing which software is installed on the network and which servers it is configured to connect to.

By understanding the design of a system and then monitoring and correlating all available data about that system, combined with conventional defences and monitoring tools, an overall picture of a network can be produced that gives defenders a better chance of protecting their network against successful attack when compared to point solutions and the potential overload of alert data, including false-positives.
5.3 The Balance of Defence

While the outlook appears bleak for the successful defence of a network, there is a balance that must be struck. On the one hand, the theoretical ideal is to purchase sufficient defensive measures that an attack never successfully compromises a network, but this is an unrealistic and potentially expensive route to take. On the other hand, not defending a network and waiting to disrupt and detect infections after they happen will lead to increased compromise, losses and impact. The other balance is between security and usability – the only truly secure computer is one that is not networked and is switched off, but this computer is not usefully usable by the business. A balance must be made between risk and the cost of compromise.

Implementing the Council on CyberSecurity Top 20 Critical Security Controls will provide a good defence against malware while maintaining a usable network. Adding the additional measures suggested in Annex B will further improve those defences against malware that uses C2 channels. Disruption of C2 channels should be performed wherever possible, and is often readily available as part of existing hardware (such as enabling outbound filtering on firewalls).

Monitoring a network can seem like an expensive process, but the intrinsic value is high. There is no way to know if a network has been compromised without knowing what is happening on it. Big data techniques can help on large networks, but smaller networks can still be monitored using similar techniques and a single, high-end device with a fast, multi-core processor and a large amount of memory. When monitoring a network then a large amount of exfiltrated data is easier to spot than low volume command channels, but both could be visible if trends are examined over the longer term.

The best defence is to separate networks, place controls at their boundaries and use thin clients, but this can be a significant architecture change for most organisations. As an alternative, organisations can invest in well-placed defences (not just focusing on servers, but also on desktops, where the compromise often occurs) and ensure that they are properly configured and that all features are being usefully utilised. Networks should be locked down to the greatest extent possible that does not unduly interfere with normal business operations, using the Council on CyberSecurity’s Top 20 Critical Security Controls as a baseline. Finally, the system should be pro-actively monitored for signs of compromise by examining trends in data and the security metrics described, rather than relying on specific alerts or waiting for a compromise to become obvious.
Having discussed the methods that could be used, based on expert knowledge and experience, the following sections describe real-world examples. Further research and references on published real-world attack examples can be found in complementary research by Birmingham University [5].

6.1 Trends in Attack

Prior to 2010 there is little mention of C2 as a characteristic of attacks in the outputs of the various referenced sources. This may be because the C2 channel was not identified as such before this date, or that its significance in the attack profile was not understood at the time.

Data from Verizon [5] seems to show an increase in the percentage of attacks in which C2 was involved, although the statistics for actual breaches (where an attack was successful and compromised a network) are slightly different.

<table>
<thead>
<tr>
<th>Year</th>
<th>C2 functionality in all recorded attacks</th>
<th>C2 functionality in breaches</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008</td>
<td>Not Reported</td>
<td>Not Reported</td>
</tr>
<tr>
<td>2009</td>
<td>Not Reported</td>
<td>Not Reported</td>
</tr>
<tr>
<td>2010</td>
<td>1%</td>
<td>11%</td>
</tr>
<tr>
<td>2011</td>
<td>9%</td>
<td>4%</td>
</tr>
<tr>
<td>2012</td>
<td>49%</td>
<td>20%</td>
</tr>
<tr>
<td>2013</td>
<td>51%</td>
<td>21%</td>
</tr>
</tbody>
</table>

Table 1 – Percentage of all attacks (successful and unsuccessful) in which C2 was used, and percentage of successful attacks in which C2 was used

There appears to be upward trends in both the number of attacks and the number of breaches that involve a C2 channel. However, caution should be exercised when reviewing these statistics as reporting methods have changed over the term of the sample, the analysis methods have matured, and the report producers have evolved the way they present their data.

There is little information available on the type of C2 channel employed in early attacks. Once again, this may be due to the relative immaturity of the data collection and analysis processes at the time. The more recent data that produced the statistics in Table 1 is richer in detail and suggests that the predominant C2 channels are IRC, HTTP and DNS. Trends are difficult to determine with such a scarcity of reliable historical data, but the initial views indicate that these channels remain popular while at the same time some practitioners are developing more sophisticated methods.

There are recent instances where email to and from an external mail account has been identified as the C2 channel. It is worthy of note that this channel was identified by inspection and deconstruction of the malware, rather than detection of the channel traffic. This is likely to be because the current detection methods were not sufficient to spot the C2 channel, but that the communication method for the channel could readily be determined once the malware had been caught and dissected.

6.2 Real-world Investigation

As well as examining what C2 is and how it can be disrupted or detected in the abstract, it is also important to understand what has been seen in the real world. The following specific examples come from the monitoring of a small set of commercial networks that was undertaken as part of this study. Further details of world-wide trends and other real-world examples are contained in the annexes.

A full copy of the network traffic was collected over a number of days on the Internet edge of the monitored networks. This data contained every packet that was transmitted from each network to any other network. As C2 communication needs to happen via a network, all such traffic is recorded in this data set (in addition to all of the legitimate traffic that was also transferred across those connections). The techniques described in the Defence by Monitoring section were then employed, presenting the data as a number of bar graphs with a column per IP address.
6.2.1 Network Analysis

An initial investigation of the networks looked at the graphs as displayed by QinetiQ’s APT Investigation Workbench – an internal prototype tool that has been developed to visualise such large data sets from monitored networks. These graphs showed several spikes in metrics for some IP addresses that appeared to be highly unusual. These spikes were seen in the comparative rarity of the Snort alerts\(^3\), the number of ICMP messages sent\(^4\) by some addresses, the number of IP ranges that some machines connected to\(^5\), and the amount of outbound traffic per IP address. Even when scaled quite small (three of the graphs are pictured in Figure 7, below), the graphs produced by the workbench clearly show when an IP address is exhibiting anomalous behaviour. Not all anomalous behaviour is malicious, but benign anomalies can always be explained. Although a spike in these graphs is not necessarily associated with C2 channels, it was found that these instances were related to malware infection that performed network reconnaissance, data exfiltration and other behaviour indicative of controlled malware.

The first graph in Figure 7 shows the rarity of Snort alerts, which has a number of hosts that are showing high scores that may indicate malware. This helps focus the search compared to an unweighted metric, but the number of high scores still provides a number of targets for investigation.

The second graph displays ICMP traffic, which shows a much smaller set of machines with unusual behaviour. The tall bar in the centre of the graph was a machine performing reconnaissance against other monitored networks. This behaviour is indicative of a targeted attack with a C2 channel, and subsequent investigation found a Trojan infection with a C2 channel for an attacker to control the activities of the machine.

The third graph shows the number of IP addresses that a machine has connected to, grouped into ranges, for a subset of the machines. As with the ICMP graph, there is a small number of obvious targets for investigation that are performing a high number of communications with an unusually wide variety of IP addresses. Some of this communication was with IP ranges that were not associated with the business activities of those networks and were found to be indicative of temporary C2 control nodes. Further Trojan infections were found.

Figure 7 - Three APT Investigation Workbench outputs, clearly showing anomalies

3. To avoid missing important issues due to false-negatives, systems can instead generate false-positives. Previous experience has shown that bad programming practice and deviation from official standards (which is nevertheless accepted as allowable by lax implementation of the standard in services) can cause such false-positives. The “rarity” metric weighted scores such that those alerts that were not frequently seen were given a higher value than those that were seen extremely frequently, thereby trying to spot the more subtle attacks that do not trigger large quantities of alert.

4. ICMP is generally used as a diagnostics protocol and most machines only rarely need to send ICMP messages. However, ICMP “ping scans” can be used to enumerate the machines on a network as part of a reconnaissance step.

5. IP addresses are broken into four sections by dots. Each section can have a value of 0 to 255 and each section provides increasing specificity of the network the machine is on. Even when connecting to websites on the Internet, only a small number of connections to a subset of all possible addresses are likely to receive connections from an individual machine. Large numbers of connections to different IP addresses across a large subset of the values of the first section can be used as a network scan or as connections to a botnet or peer-to-peer network.
The data was also processed using a variety of rule sets, including rules for Snort targeted at detecting emerging threats. The first attack that was identified was a set of C2 check-in commands from an instance of the DoFoil Trojan on a single IP address. This corroborated the conclusion, from other analysis of the pattern of the machine’s behaviour, that the machine was compromised. Further investigation tracked the communication to domains made of random characters that had previously been identified as unusual. Other Trojans were also discovered by the rule set, but the mix of one thousand or so alerts about different Trojans were swamped by fifteen times that number of alerts warning about a possible Trojan infection based on inbound Microsoft Terminal Service connections or other anomalous behaviour. This shows that purely relying on alerts from an intrusion detection system, as opposed to metrics taken over a longer period, can lead to analysis overload. A preliminary examination of the major Trojan targets indicated that they each also displayed anomalous behaviour against the other metrics.

### 6.2.2 C2 packets

The network analysis that was performed detected multiple Trojans and indicators of other malware infections on several machines. The Snort rules also identified the C2 packets of a number of well-known Trojans as part of an “emerging threats” rule set, where the Trojan has been expertly analysed and reverse engineered to understand its behaviour so as to enable the production of appropriate rules for detection tools. While these packets were detected, and the behaviour associated with C2 channels was identified, the packets themselves were difficult to discover without detailed knowledge of the Trojan.

An example of the difficulty faced in identifying C2 channels is shown in McAfee’s presentation on “Taming Worms, RATS and Dragons” [7]. This presentation dissects a number of malware programs, including Gh0st RAT (Remote Access Trojan) and its C2 channel. The C2 channel uses encoding to compress and obfuscate its command channel, and the look-up tables for some of this data are shown. Without these tables the data is indecipherable, even if it is sent unencrypted. Once the malware and its traffic is dissected by professionals at major malware labs such as McAfee’s then patterns and signatures may be found that can be detected by network defences, but these patterns will not be generic across all C2 channels and are not known until after extensive expert analysis.

Figure 8, below, shows a packet that was identified by Snort rules as being a check-in packet for a Trojan, which is part of the C2 process where the infected machine reports that it is available to be controlled and exploited. The data within this packet is the string of numbers at the end, which follows on from a correctly formed set of HTTP headers (including an accurately faked Internet Explorer 7 user agent string). This packet cannot be picked out based on its headers. The content of the packet is indecipherable without knowledge of the internal coding mechanism of the C2 channel. The content is also not easily distinguishable from any binary file format or encrypted data, as it is just a string of bits. This is an intentional design of C2 channels to disguise their behaviour. The only giveaway feature of this request is that it is being sent to the domain ggacsev34nwas.us, which is a meaningless string that is unlikely to be part of a legitimate request. If a better choice of domain name had been made then there would be no distinguishing features that would generically identify this as a C2 packet.

Figure 8 - Example C2 check-in packet from Dofoil Trojan

Figure 9, below, shows a legitimate packet collected from the same network. In contrast to the C2 packet, the content of this packet has been recognised as a string and can be read in clear text on the right-hand side of the figure (truncated for brevity). This is because Google are not trying to hide the behaviour of their update process and are using XML as a means of encoding the data that their system needs to use to provide the correct update and record appropriate metrics.
6.2.3 Investigation conclusions

Although specific C2 traffic was discovered from infection by known Trojans, no advanced C2 channels were discovered that might be expected from an APT. This may be due to the other Trojan activity masking their behaviour or it may be due to a lack of APTs, but the examples in the previous section show the difficulty of discovering the unknown when it is designed to remain hidden. The lack of proof of APTs does not prove a lack of APT infection and advanced C2 channel, since the intention of APTs is to be sufficiently advanced and well designed as to remain hidden on the network.

As all C2 traffic appeared to be related to known Trojans then the communication was not extensively disassembled, since other researchers will have already done this. Further outbound traffic patterns that appeared to be suspicious could not be suitably analysed within the constraints of the data collection agreement and so it could not be determined whether these were full C2 channels, automated data exfiltration, or merely unexpected traffic (either legitimate but unknown traffic or traffic that should not have been on the system but that was from legitimate applications).

In general, although the Trojans were identified by the Emerging Threats rules and could have been blocked by existing defences, the patterns of behaviour of nodes that were infected was more obvious to find in amongst the noise of other alerts than just viewing the alerts themselves. This supports the conclusion that although some malware and C2 channels could be detected by technical defences, the more consistent indicators of compromise were the patterns of their behaviour.
7 Conclusion

This report has provided an introduction to malware and the current “state of the nation” with regards to computer networks. This was then followed by a description of what Command and Control (C2) is and how it is used. Having described how malware authors create C2 channels, the document described methods of detecting and disrupting C2, before describing some real-world findings.

The preliminary assumption that must be made in the modern networked world is that the network is not necessarily clean. Commodity malware attacks are worryingly prevalent, and Advanced Persistent Threats (APTs) could be on a network without having been detected. Without this assumption then any future inferences based on monitored data may be incorrect as they’re based on an inaccurate baseline.

C2 channels (and especially C2 for APTs) can be difficult to identify by micro-inspection of packets and protocols, and are therefore extremely difficult to prevent. This is because alerting on specific patterns tends to generate too many false-positives to be useful. However, observation of the behaviour of groups of nodes over extended periods can spot activities such as network reconnaissance and data exfiltration. A good analogy is the detection and prevention of organised crime, where communication between key targets is often not the first indicator that is found, but becomes useful to identify the scale of the problem once the impact of the illegal activities (such as smuggling, murder and fraud) are detected. In these cases, detecting and disrupting the communications channels initially is difficult or impractical, but becomes useful later.

Depending on the technique used, there are still best practices that can be used to defend against malware in general and at least a proportion of C2 channels. All networks should follow best practice, as laid out in the Council on CyberSecurity’s Critical Security Controls list, as a minimum. Some techniques, such as proxy servers and protocol anomaly detection, may detect or block a subset of C2 channels, but the authors of more advanced malware are already using Internet-based services and protocols that may be expected to be seen on a network, thereby hiding their attacks amongst legitimate behaviour. Additional measures beyond best practice can further secure a network against malware and C2 but a key defence is the use of monitoring to detect anomalous behaviour, which an IDS such as Snort can be part of with an emerging threat rules set or GovCert rules set.

Once monitoring data is collected then it needs to be viewed by skilled analysts who know the network and understand what traffic should be seen and which services should be running on which servers. With this knowledge and a holistic view of the network activity, malware compromise can be detected more easily than C2 channels themselves in the same way that the ripples on a pond are easier to spot than the feeding fish or the dropped rock that caused them. Due to the volumes of data that need to be collected, it will generally be necessary to use Big Data storage and processing tools to appropriately handle this data.

Finally, the battle against C2 channels is an on-going arms race between attackers and defenders, the same as the battle against malware in general. Where C2 channels are spotted on networks then future malware may be written to use new techniques, or pre-existing but previously unused techniques may deployed. A clever malware developer will not use all of their novel “crown jewels” techniques when a simpler technique will suffice, except when targeting the most valuable assets. As with other security, the only way to ensure a network is completely proof against these threats is to secure it to such an extent that it is no longer useful for business operations. This is obviously not feasible, and so a balance must be struck between ideal security and business operations.
## 8 List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>APT</td>
<td>Advanced Persistent Threat</td>
</tr>
<tr>
<td>AV</td>
<td>Anti-Virus</td>
</tr>
<tr>
<td>BYOD</td>
<td>Bring Your Own Device</td>
</tr>
<tr>
<td>C2</td>
<td>Command and Control</td>
</tr>
<tr>
<td>DGA</td>
<td>Domain Generation Algorithm</td>
</tr>
<tr>
<td>DMZ</td>
<td>Demilitarized Zone</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name Service</td>
</tr>
<tr>
<td>HIDS</td>
<td>Host Intrusion Detection System</td>
</tr>
<tr>
<td>HIPS</td>
<td>Host Intrusion Prevention System</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>ICMP</td>
<td>Internet Control Message Protocol</td>
</tr>
<tr>
<td>IDS</td>
<td>Intrusion Detection System</td>
</tr>
<tr>
<td>IPS</td>
<td>Intrusion Prevention System</td>
</tr>
<tr>
<td>IPv6</td>
<td>Internet Protocol version 6</td>
</tr>
<tr>
<td>IRC</td>
<td>Internet Relay Chat</td>
</tr>
<tr>
<td>NAT</td>
<td>Network Address Translation</td>
</tr>
<tr>
<td>P2P</td>
<td>Peer-to-Peer</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Sockets Layer</td>
</tr>
<tr>
<td>TBLH</td>
<td>Take a Business Laptop Home</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
<tr>
<td>VoIP</td>
<td>Voice over IP</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
<tr>
<td>XMPP</td>
<td>Extensible Messaging and Presence Protocol (formerly Jabber)</td>
</tr>
</tbody>
</table>
9 Bibliography

The following references are all correct and accessible at the time of writing.


Annex A Command & Control Channels

The following sections provide a penetration testing professional’s view of what a C2 channel is, why they’re used, and how they work, as well as some of the constraints that are faced by C2 programmers. The emphasis here is on APTs – a subset of malware that uses particularly advanced C2 channels – as they pose the highest risk and are where new C2 channels are often first seen. All of these attacks will generally arrive in common, generic malware over time, though.

A.1 Establishing a Command & Control Channel

In spite of existing security controls, we live in a world where corporate systems do get compromised, often through previously unexpected or unanticipated attack paths. This complex, novel, subtle and unexpected type of malware infection is often termed as an Advanced Persistent Threat (APT) due to the method of delivery, exploitation and unknown malicious intent of the payload. The goal of security in such a world cannot realistically be to prevent 100% of all infections, but instead it should be to accept the inevitable and to be prepared to deal with it using appropriate disinfection and recovery procedures.

Given that an APT can get onto a corporate network undetected, how do we detect and eliminate the threat? Lockheed Martin [4] performed an analysis of the steps typically taken by an attacker and designed their Cyber Kill Chain™. It defines seven steps where action can be taken to detect and terminate an APT. Step six of their kill chain is the Command and Control (C2) channel, and whilst they do give a number of examples of APT infections that they have detected and the C2 channel used, they do not define a generic set of procedures that could be used to detect C2 channels.

In order to try and detect an APT C2 channel, we must first understand why an APT needs to communicate on a network and what type of information it might be sending.

A.2 What is a C2 channel?

The C2 channel utilised by the malware will depend upon two factors. Firstly, it will depend upon the design choice of the attacker. The attacker will need to determine how often they need to communicate with their malware in order to achieve their objective. Secondly, if the attack is an APT then it will need to be able to exploit the network architecture surrounding its host system and take advantage of the outbound communications channels available. Alternatively, if the malware is a Trojan or BotNet then it will need to be commanded to undertake specific activities and attacks. Consequently, malware may be pre-programmed to take advantage of one or more of C2 channels – the types of channels exploited will be discussed later in this paper.

APTs are often delivered to networks where the internal network structure is unknown to the attacker. It would be extremely challenging for the best attackers or nation state actors to design an autonomous piece of malicious software that can achieve a specific goal or hunt down specific information on a network which the attacker may have little to no knowledge of. Therefore, the vast majority of APTs will require some form of guidance from a human operator or master. If the APT can communicate with its master then the attacker will have effectively gained user level access to the compromised workstation, but remotely. The C2 channel will allow the remote attacker to run the same commands as a regular system user and retrieve the results of those commands.

Regardless of the channel used, C2 channels in malware tend to follow a similar traffic pattern.

1. Registration of a new infection/Resumption existing infection session
2. Transfer of command(s) to be executed
3. Exfiltration of results from any command(s) executed

Phase 1 of an APT communications model will be the point at which the malware will make its initial call home to its master. Based upon the analysis QinetiQ has performed upon a number of malware samples, an APT will perform an initial assessment of an infected host, extracting and sending information to the attacker. The type of information that is typically extracted includes the host-name, domain name, IP address and networking information, and the user-name of the user who has contracted the APT infection. The attacker can use this information to determine if the APT is on an existing host, or if it has come from a newly infected host.

Phase 2 of the communications model is when the malware will be instructed by the attacker to execute a command. Typically, commands fall into three main categories:

Execute – This command will allow the attacker to run an operating system command or to execute a program or script on a compromised system, or a custom command which is embedded into the APT.
Download – This command gives the attacker the ability...
to instruct the APT to download a file from a source of the attackers choosing. Typically, the types of files downloaded are additional attack tools that can be used to extract passwords or escalate privileges on the infected host. Sometimes this command is used to replace an APT with a newer version that is better suited to its environment.

Upload – This command is used by the attacker to instruct the APT to upload the contents of a file to a location of the attackers choosing.

Phase 3 of the model is when the APT reports back any information that it has collected. Typically this will be the output of any commands executed in phase 2. This information may be sent in clear text or encoded to prevent detection. The information may not necessarily be sent to the same server as the commands collected from phase 2. Once phase 3 has completed, the APT will loop back to phase 2.

The C2 channel is a critical component of any sophisticated APT. Using only the simple command set described above, it is likely that an attacker will be able to fully compromise an APT affected host, and then compromise other hosts on the networks in which it resides. All that the attacker needs is the time. Once the C2 channel has been established, the race is on to detect and disrupt it before the attacker can spread through the network and achieve their objective.

A.3 C2 Communications Model

The C2 communications model used by the APT will depend upon how stealthy the attacker wishes to remain, and how much work (in terms of intellectual property) they are willing to lose should they be detected. If an attacker’s most advanced C2 channel is thwarted, it may lead to its detection upon other systems that the attacker controls. Attackers will also choose a communications model based upon the value of their target and the perceived difficulty of maintaining a compromise in that environment. In general there are two communications models commonly employed by APTs in the wild. These models are dictated by the flow of network traffic between the APT and the C2 server and are as follow:

A.3.1 Push Model

The first type of APT will attempt to make a direct connection back to its master, where the master can send commands directly to the APT for execution. This connection is often made using a full TCP connection back to a command and control server and it is used to inform the attacker that a successful infection has taken place, to obtain commands to be run, and to return the results of any previously run commands. Command collection and execution happen instantaneously almost as if the attacker was present at the keyboard of the infected host. This model typically relies upon the Internet Relay Chat (IRC) protocol as the bearer, and infected hosts join channels on an IRC server and/or channel under the attackers control. Malware analysts [8] describe this method as the ‘Push Model’ as commands are pushed directly to the APT to be executed.

A.3.2 Pull Model

A more sophisticated APT model uses a drop-box style communications model. The APT will attempt to contact its master indirectly using a stateless protocol, picking up message left by the attacker and depositing any information requested. A typical example of this uses HTTP communications and issues GET requests to a website under the control of the attacker to collect commands that the attacker would like to be executed. In return POST requests are used to submit the output of those commands back to the attacker’s website. The attacker can recover the results from their web server logs. Due to the nature of this type of protocol, the APT will need to periodically check for new commands to run, meaning that there is likely to be a delay in commands being retrieved, executed and its results returned. This is commonly known as the ‘Pull model’ as the APT polls a C2 server for new commands at its discretion.

A.3.3 The Autonomous APT

Whilst there are two communications models that can be used, some of the most sophisticated (but highly targeted) APTs can be fully or partially autonomous. These are used to achieve a very specific effect on network infrastructure that is air-gapped and inaccessible from the Internet. The Stuxnet worm [9] is a particularly good example of this as: whilst equipped with an HTTP communications module, it also spread autonomously via USB keys, exploited zero-day vulnerabilities and was also armed with a payload designed to subtly sabotage the operating parameters of an industrial centrifuge. For the purposes of this paper, we will only be examining the malware and APTs which make some form of network communications with master servers, i.e. the Push and Pull models.
A.3.4 C2 Servers

An important part of Command and Control, at least in the initial phases of infection, is the C2 server infrastructure. This will be a host, or set of hosts, which the attacker already has control over and are used to orchestrate their APT infected hosts. For redundancy, most attackers will operate a number of C2 servers. C2 servers are often hosts which have been compromised and re-purposed by the attacker. By using multiple C2 servers, the attacker will still be able to contact and control their compromised hosts even if some of their C2 servers are taken down. Redundancy permits the attacker to maintain access to their compromised hosts.

There are some exceptions to this traditional C2 server model. For example, the ZeroAccess malware which some have estimated has infected millions of hosts [9] worldwide and uses a Peer to Peer (P2P) model to remove the requirement for centralized C2 servers once hosts have been added to the P2P network. Some ZeroAccess infected nodes are promoted to super-nodes which are used to relay command and control messages to other infected hosts. By removing the central C2 server requirement, it becomes more difficult to deny access to the C2 servers using conventional firewalls. However, the techniques required to detect and disrupt this kind of distributed command and control channel remain the same.

A.4 Network Architecture Constraints

The choice of communication model used by a piece of malware will be constrained by the network architecture surrounding its compromised host. This network architecture will largely depend upon the type of organisation, its size, and of course their budget. For example, an SME is more likely to use a relatively simple network configuration with a few simple firewalls and anti-virus installed on user workstations. The goal of this type of network is usually to provide functionality over security, and is designed to allow employees to get on with their job.

On the other hand, a large government department is more likely to have an extensive IT budget and a clearly defined IT security strategy. They are also more likely to have large segmented networks split over a number of physical locations, with functional components logically separated from one another. They are also more likely to have installed layers of defensive measures, perform network monitoring and filter all incoming and outgoing network traffic.

Most organisations understand and apply basic security principles to ensure that hosts are patched and kept up-to-date, with Internet facing hosts contained within a DMZ environment. However, the security of the user community is often overlooked. Most organisations will require their employees to access Internet services like web and email and can become over-reliant upon the security appliances, policies and procedures that they have put in place.

Security appliances and network architecture will place controls and limitations upon the protocols and classes of traffic that are able to enter and leave a network, thereby affecting the choice of C2 channel. The first line of defence in most organisations is their Internet-facing firewalls. These are often sold to organisations as perimeter defences used protect their internal networks from external attack. However, organisations do not always consider the outbound network traffic leaving their networks, and only configure their firewalls to block incoming traffic.

The second line of defence that organisations often implement are email attachment content checkers and AV scanners. The ability to collect and validate email before it is delivered gives an organisation the ability to prevent the spread of known malicious content. That said, the files that are often permitted through such devices, such as PDFs and office documents, are the very formats targeted by attackers to deliver exploits and payloads. Furthermore, any AV product integrated into such appliances will only be as effective as their signature database and heuristics.

Web proxies are often the third line of defence that is implemented. Initially designed to perform web-caching, they have evolved into web filtering devices which can be used to control what content a user community can access. Again, AV products are often incorporated into these devices, offering protection against known malicious downloadable material. However, these devices are only as good as their configuration. Often the product vendor is trusted to categorise web content, effectively giving them control of what content can be viewed within an organisation. Furthermore, the impact of social networking has changed the way many organisations do their business and such sites are often accessible from the workplace.
The fourth line of defence for an organisation is the user workstation configuration and any security software installed on it. Again, this domain is dominated by AV products and, optionally, host-intrusion detection systems (HIDS). HIDS systems aim to detect and prevent exploitation, and can be configured to detect suspicious process level behaviour. This layer of defence aims to catch anything that might have made it past any of the previous defences and actually reached a host that could be exploited.

Network Intrusion Detection Systems (IDS) are the final line of defence that is included in a secure architecture. These systems perform analysis of network traffic flows and use signatures to detect known malicious content. Again, like most AV products, if the malicious content is not known then no signature will exist. In the absence of signatures, some people hope to detect threats using the heuristic models built into some IDS products. Reliance upon heuristics to detect such threats in IDS products will in all likelihood raise the false-positive detection rate. Without reliable detection, it becomes unreasonable to reliably detect real attacks that may be hidden amongst the noise of the false-positives.

A.5 Primer - The Role of DNS

The first, and possibly the most important protocol that almost all APTs and malware rely upon is DNS. Unless an APT is hard-coded with the IP addresses of its C2 servers, it will need to use DNS in order to find those IP addresses via A or AAAA record (the DNS records which map human-friendly internet names to IP addresses). Reliance upon a hard-coded set of IP addresses tends to be avoided in an APT, as it is trivial to identify these and block access to known C2 server IP addresses with firewall rules. Just as DNS was introduced to simplify a difficult to remember numbering system to a human-friendly and readable format, so too it is used by attackers to add an abstraction layer between an APT and the C2 servers they control.

DNS gives the attacker the flexibility of updating the location of their C2 server should it need to change - for instance if one of the attackers C2 servers is discovered and taken down. In recent years, some malware has advanced its use of DNS to use algorithms which generate a domain name that when resolved can be used to find a C2 server or entry point into a P2P network. Known as Domain Generation Algorithm (DGA) [11], the algorithm will also be known by the attacker and used to generate and register a set of domains as required, as well as create DNS entries for it, in time for the malware to find its C2 server. Using this technique allows the attacker to minimise the amount of time they expose the location of a C2 servers via DNS, making it more difficult for defence measures to be applied in a timely fashion.

Once an APT has managed to locate its C2 server using DNS or otherwise, it will attempt to contact it using one or more of the protocols described below. The protocols listed below are intended to illustrate the types of protocols that may be able to egress a network. It is important to realise that this list is not exhaustive and does not detail custom protocols invented by the attackers. However, these protocols are those most often used as a bearer for an attacker’s communications.

A.6 Current Protocol Selection and Threats

This section details the protocols that are often used by current APTs as a bearer for their C2 channel. The C2 channel used by an APT will be designed to exploit weakly configured network architectures. If we look at the defences illustrated in the previous section, we can see how attackers have evolved their C2 channels to escape the confines of a well-controlled network.

A.6.1 The IRC Protocol

Internet Relay Chat (IRC) has been a popular choice for malware authors as the C2 channel for a number of years since 1999 [12] when the PrettyPark worm first surfaced. Originally designed as chat room software, IRC provides a convenient mechanism for an attacker to control compromised hosts via a chat-room style user interface. IRC supports one-to-one communications between individual users, or group chat features within attacker controlled channels, and it also implements simple mechanisms for the channel owner to control who can access their channel.

The IRC protocol fits the attackers’ needs in several ways. Firstly, by using a simple password protection mechanism implemented in a channel bot, the attacker can prevent unauthorised users from interacting with their infected hosts. Secondly, IRC channels are a convenient way to send group commands to multiple
infected hosts in an instant. Lastly, the one-to-one private messaging system allows the attacker to push commands to specific infected hosts and receive messages from individual infected hosts.

Mapping the IRC protocol to the three phase communications model described earlier, we often see that in phase 1, an APT will join a pre-programmed IRC channel on one or more servers controlled by the attacker. As each user on an IRC network needs a unique nickname, the APT will attempt to generate one from information it can gather from the infected host as previously described. Typically, the nickname will be a digest of the host-name, domain-name and any other information obtained from the environmental variables stored on the infected host. This dynamic nickname allows the APT to generate a consistent identifier for an infected host, making it distinct from other infected hosts and allowing an attacker to track a specific host in the event that a host is powered down or temporarily removed from its network. IRC is a connection oriented TCP protocol and as such phases 2 and 3 of the C2 communications model are represented as either side of this connection.

To establish a connection to its C2 server, an infected host would need to be located in a network environment with little or no egress filtering towards the Internet. Therefore, IRC is often the easiest of C2 channels. Although IRC uses a clear text protocol, many APTs have been designed to encrypt or encode communications over IRC. By encrypting the contents of messages, the attacker can prevent eavesdroppers (malware analysts/IDS) from analysing the traffic between their server and their infected hosts, potentially prolonging their grasp on an infected host.

More advanced forms of this C2 channel are required when the target environment has employed egress firewalls that block outbound access to typical IRC TCP ports, such as 6667. To counter this, some attackers host their IRC server on alternative ports, such as ports 80 or 443, which are often permitted for outbound web traffic.

QinetiQ has analysed a number of malware samples which attempt to make an initial connection back to its C2 server over TCP ports 80. Some examples attempted to make connections over TCP port 443; however they did not attempt to use SSL for that connection and simply made direct connections to an IRC service hosted on this port. It is important to note that, even though TCP ports 80 and 443 were used, the network traffic was still raw IRC protocol. Some IDS systems will detect network activity of this type as a protocol anomaly. Some networks will have no reason to run IRC at all, and if its presence is detected then it should be taken seriously.

A.6.1.1 Variations on IRC - Instant Messaging

Instant Messaging (IM) clients have gained popularity within organisations in recent years, allowing staff to message and interact with colleagues. If allowed outbound, IM becomes an attractive mechanism way of forming the C2 channel. Just like IRC before it, IM provides a quick and efficient mechanism to push commands to infected hosts and to retrieve the results. There are many IM protocols around including:

- Skype
- Jabber/ Extensible Messaging and Presence Protocol (XMPP)/Google Talk
- Session Initiation Protocol (SIP)

Skype and Google Talk are interesting to an attacker because of the added encryption which will make the analysis of network traffic generated by their APT very difficult. Protocols like Skype make use of both TCP and UDP and a variety of ports, which makes detecting and successfully blocking it difficult for a network administrator.

Organisations need to be aware of the IM protocols that they allow out of their networks and prepare for any that they allow to be abused by an attacker.

A.6.2 Peer to Peer Protocols

Peer to Peer (P2P) networking came to the forefront in 1999 with the Napster program when it became popular for sharing digital content, legal and illegal, over the Internet. As its name suggests, its architecture is distributed in nature with hosts connected directly to other hosts. Information is relayed directly between these hosts and the need for a centralized orchestrating infrastructure is minimal. P2P is used to connect hosts together in order to form a private network that is overlaid on-top of another network, in most cases this is the Internet. The machines in this private network are called a “herd”. To improve the efficiency of data transfer, most P2P networks incorporate an internal routing mechanism to allow messages to be routed to their destinations efficiently.
More recently, botnet operators have taken advantage of peering to remove the need for dedicated C2 servers. They instead, use the interconnectedness of their botnet as the C2 channel and let infected hosts route C2 messages and their results through the P2P network.

Whilst advanced, P2P architectures are not best suited to corporate environments. They rely upon poor egress filtering, and poor ingress filtering – often taking advantage of protocols such as UPnP to forward Internet facing TCP and UDP ports to infected hosts. As such, the current generation of P2P malware tends to infect home networks. For example, the ZeroAccess Trojan has made use of TCP and UDP ports [12] as part of its C2 channel and has been used to infect millions of home computers in the USA and UK. Instead of being used to steal secrets, it appears to be used to harvest processing resources of infected hosts to generate Bitcoins and to perform click-fraud to earn money for its master.

The unique architecture and many-to-many communications model use by P2P can still be mapped to the C2 communications model described earlier. For example, the P2P malware will still need to find a herd to join, and will most likely use DNS to find its entry point to the P2P network. The registration phase will involve the creation of a dynamically generated identifier before making a connection made to the host resolved via DNS. Once connected, the malware will register itself (phase 1) as an infected host, share its unique identifier, and proceed to download a list of other infected hosts. The malware will attempt to connect to a number of its peers. Once connected to its peer network, the malware can received directed or broadcast commands (phase2) from its peers. The results of any commands executed can be reported back to its peers (phase 3) – the internal routing structure of the P2P network will ensure that the results will reach a host under the attacker’s control. Many P2P networks use the uniquely generated identifier in a Distributed Hash Table (DHT) that is shared between hosts and used for message routing purposes to remove any further reliance upon a centralised C2 server.

Whilst the current generations of P2P malware appear to be targeting home networks, they will eventually find their way into organisations as the communications techniques are developed and refined. Some P2P malware have been known to update their communications protocols [13], to adapt to their environment and become more resilient. To adapt to use within an organisation, it is likely that P2P APTs will need switch from using custom binary protocols to use some of the other communications protocols used within this paper to bypass firewall restrictions and other security devices.

### A.6.3 DNS as a tunnel

DNS is a lookup system that can be used, amongst several other functions, to map human-friendly domain names into computer usable IP addresses. The hierarchical nature of the DNS protocol makes it attractive to an attacker, as it can be used to establish a direct communications channel from deep within a target environment. There have been a number of well documented cases [14] of DNS being used to escape the confines of networks which have been hardened against attack. However, DNS is still often overlooked as a communications channel by many.

Standard DNS lookup requests use a UDP based protocol and are sent from a host to its nominated DNS server. Clients will send questions for the server to answer. In a human-readable format this may look like “What is the IP address for www.google.com?” If the DNS server knows the answer, then it will respond with “The IP address of www.google.com is 173.194.41.98 or 173.194.41.96 or <and so on>”. If the nominated DNS server does not know the answer, it will forward the request onto another DNS server higher up in the hierarchy until it an answer is found from an authoritative DNS server or the request times-out because no answer could be found. DNS can also return other records types such textual records (TXT) which contain whatever text the DNS server operator wishes. Users have control over the address which is sent to a DNS server, and this address can contain up to 255 octets of data, which in reality equates to 253 ASCII characters. Similarly DNS responses have the same size limitations and can contain multiple answers allowing a response to fill a DNS UDP packet up to its hard limit of 512 octets. We can see that there is enough space within DNS questions and answers to transmit data in and use it as a C2 channel, all that the attacker requires is an authoritative DNS server for their domain.

In organisations that allow DNS to be resolved externally, an attacker can abuse the DNS hierarchy and DNS queries to tunnel information from within a network out to a host they control. For this example, assume that the attacker controls the domain attacker.com and its DNS server. Their APT can use DNS tunnelling to make a request for contentsofafile. attacker.com and use DNS requests to leak the contents of files on an infected host. Any responses to the DNS request may contain an answer section that can be used
to send commands for the APT to execute. This means that DNS requests can be used to provide phase 3 of the C2 communications model and DNS answers can be used to transmit commands for execution as described by phase 2. The registration phase, phase 1, for a DNS tunnel may be limited to tracking the source IP address of the DNS request or by using the transaction ID values stored within a DNS request. Alternatively, a unique identifier can be prefixed to DNS names to track individuals APT infections.

A common feature shared by DNS tunnels is that the name lookups used for DNS requests, whilst using ASCII characters, are not in any meaningful language or format. For example, data is often simply hex encoded resulting in long, non-human readable domain name requests such as e542998da234b3de2dfa.attacker.com. Additionally, in order to exfiltrate large amounts of data from a network then an equally large numbers of DNS requests will need to be made to carry the fragments of that data. Furthermore, the size of the DNS packets being generated is likely to be larger than the average DNS packets to take full advantage of the outbound bandwidth available with minimal waste due to the ratio of content to overhead.

A.6.4 Web Traffic – HTTP/HTTPS

In environments with tightly controlled egress of network traffic to the Internet, web traffic is still often allowed outbound either directly or via a web proxy. In these circumstances, HTTP can be used by an APT as a bearer to establish the C2 channel. HTTP is a simple textual based protocol which supports variable length requests and responses and is the protocol most commonly used by most web browsers. A simple web request that retrieves a test page from the BBC website using the Google Chrome browser may look like:

```http
GET /test HTTP/1.1
Host: www.bbc.co.uk
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/30.0.1599.114 Safari/537.36
DNT: 1
Accept-Encoding: gzip,deflate,sdch
```

Before a request is sent, a DNS request will be made to establish the IP address of the server which holds the desired page. The method used in this instance was a GET request, but there are other HTTP verbs that can be used to access data on a website the next most popular being a POST request. POST requests are used to send large amounts of data to a website, and are normally used when a user submits a web form. The following is a typical response to the above request:

HTTP/1.0 200 OK
Date: Mon, 04 Nov 2013 13:34:57 GMT
Server: Apache
Accept-Ranges: bytes
Content-Length: 214
Content-Type: text/html
<html>
<head>
<title>BBC - test</title>
<style>body{font-family:arial,sans-serif;}</style>
</head>
<body>
<p>This is a test site</p>
</body>
</html>

The response from the web server contains two sections, the first section being the header and the last section is the body of the actual response. When this protocol is used by an APT, the simplest mechanism available is to use GET requests to retrieve commands from a web-server under the control of the attacker and to use POST requests to return the results of any commands run by the APT. Unlike the previously described protocols, the use of HTTP uses the Pull model of communication. A timer is often used to periodically request new commands from the attacker controlled web-server.

When compared to the C2 communications model defined earlier, the registration phase will often be observed as the first request to the attacker’s web-server. The APT will upload information about the infected host in exchange for a session token. This token is included with all subsequent requests to the attacker’s servers, allowing the tracking of individual infected hosts. Phase 2 will occur periodically when the APT requests a command from the attacker; this might be through a GET requests for a specific file on the attacker’s web-server, such as /command.txt. Reporting the results of this command in phase 3 will happen...
through a POST request back to the attacker’s web-server, thereby completing the cycle.

HTTP offers the attacker flexibility in where they store their commands and responses. Apart from the more obvious locations such as the parameters sent with GET and POST requests, there are a number of HTTP headers that can be used to carry data between a client and server which may not be immediately attract attention should the network traffic be analysed. To make use of the textual fields of an HTTP request or response, the attacker will typically have to encode any files or data that they send. Base64 or hex encoding is commonly used to convert binary data into ASCII printable characters suitable for use with HTTP, and is symptomatic of data smuggling. There are many examples of Base64 encoding or being used to tunnel data in the Malware Traffic Patterns spread-sheet shared by DeependResearch.org [16].

QinetiQ has observed that many of the web-servers being used as C2 servers are genuine web-servers that appear to have been compromised and are being utilised by the attacker. In these cases, we often see web services bound to alternative ports that are also commonly used for web services such as TCP ports 443, 8000, 8080, and 8443. In order to evade detection by network IDS, it is possible for the attacker to use HTTP over SSL (HTTPS) to encrypt all of the traffic between the APT host and the C2 server. Even through ports commonly used for SSL encrypted web traffic are being used, not all communications to those ports is encrypted. QinetiQ has analysed malware samples which attempt to make clear text HTTP communications over TCP port 443, which is normally used for encrypted HTTPS traffic.

Other well-known examples of HTTP-based C2 channels include DropSmack [17] which uses the DropBox service to create a C2 channel and tunnel files in and out of an organisation, and websites with user contributions. This latter technique is increasingly employed to use social networking sites to host the commands for an APT to execute in phase 2 of the C2 communications model. Results can also be posted back to other social network sites such as blogs, Twitter, Facebook or other public sites. For example, there are examples of APTs which use Twitter to retrieve commands [18] [19]. Others have been found that use Google Documents [20] as a way to tunnel commands back to a C2 server. Some malware have been shown to use amazon.com reviews to post back the results of commands that have been executed. The use of sites which allow user-submitted content, such as social networking, to control an APT is novel and interesting as it makes the blocking of such command channels very difficult – some organisations may block access to Twitter, but how many will block access to Google? With an increased dependency upon social media for marketing, and existing acceptance of its casual use by staff during breaks, it is likely that these will be exploited as C2 channels for some time to come. Indeed, it becomes a challenge for the operators of such sites to become more resilient to this usage profile. Additionally, we are starting to see APTs which use a blended approach, and use information scattered across a number of popular sites to form a C2 channel.

Another popular evasion technique is the use of steganography, the ability to hide or encode secret information within another type of information. The key to extracting that information will be known only to the attacker and their APT. Steganography provides a layer of obfuscation in the traffic flows between their infected host and their C2 server, and makes the detection by humans very difficult. Typical examples of steganography include hiding data within images such as JPEGs or GIFs. By subtly modifying the contents of a known set of pixels within an image like a GIF or PNG, or by altering the values used to generate an image such as a JPEG, it is possible to send secret messages between an APT and the C2 server. Steganography and other less sophisticated attacks can also be used to hide information within Word documents, PDFs and other file types that may be allowed through a filtering web proxy. Just as before, if a specific file type is permitted to cross a security boundary like a web proxy, then it is possible that it may not only conceal an exploit but it could equally be used to harbour data for use as a C2 channel.

A.6.5 IPv6 and ICMP
IPv6 is the up-and-coming replacement for IPv4, and aims to add capacity to the Internet’s numerical addressing schema. Many modern operating systems come preconfigured to use IPv6 by default, but just how security appliances deal with IPv6 is not entirely clear. Organisations that already have tight controls over their IPv4 network traffic flows may not necessarily have those same controls over the IPv6 address space. With technologies such as Teredo [20] extending IPv6 onto IPv4 networks, some organisations may already be exposing internal hosts to the Internet. Indeed there are several pieces of malware [21] which take advantage of IPv6 tunnelling to communication directly to infected hosts just as they would have with an IPv4 host. IPv6 can be used to carry any of the protocols described in this paper, including IRC, HTTP and VOIP traffic.
Organisations that would have once blocked protocols like Internet Control Message Protocol (ICMP) at their perimeter may now be opening their firewalls to allow such traffic through. Some ICMP messages are required [22] to allow the normal operation of IPv6 networks, however if misconfigured, there are a variety of different messages types that may be attractive to an attacker. ICMP message are commonplace on most networks, and in small numbers they don’t tend arouse suspicion as it is often used throughout networks for diagnostic purposes. For example the ICMP echo request and ICMP echo reply messages are used by the ping command.

There have been instances [24] of malware using ICMP packets to communicate back to C2 servers, and there are many tools [25] [26] available which make use of the slack space at the end of ICMP messages to send covert messages and form a communications tunnel. In order to be use ICMP as a C2 channel, the APT will need to pull commands from its C2 server using an ICMP echo request and the command to be executed will be stored in resultant ICMP echo reply message. In the same way, ICMP echo requests can be used to perform the registration phase and to return the results of any commands that have been executed.

A.6.6 BYOD

Bring Your Own Device or BYOD, whilst not a network protocol, brings a number of challenges to corporate environments. Allowing employees to attach personal devices to corporate infrastructure introduces additional channels that may be exploited by an APT to form a Pull model C2 channel. Popular BYOD devices include iPads, Android tablets and personal laptops, most of which have large amounts of built in storage which, if compromised, can be abused by an attacker to copy files and data from a secure network and then later released when that device is connected to a less secure network.

BYOD really is the modern Trojan horse, and can be used to side-step all conventional security boundaries installed on a corporate network, effectively extending the corporate security perimeter to include that of the least secure network used. At this point the attacker can use any protocol they wish to form the C2 channel and upload a set of commands to be executed once the compromised device has been connected to the target environment.

The same can be said for corporate assets that employees are expected to use from their home or mobile environments. If a corporate laptop is poorly configured and routinely connected to uncontrolled networks, then at this point a C2 channel can be established and a command set uploaded to the APT in the same way as BYOD.

A.7 Future Protocol Selection and Threats

While all of the channels described already have been seen being used in malware “in the wild”, they are not an exhaustive list of C2 channel methods. Whenever a protocol or technology is used on a network then attackers will look for a way to exploit it. The following protocols and threats describe C2 methods that have been demonstrated in a lab or suggested theoretically but which haven’t been confirmed as having been seen in real malware.

A.7.1 VOIP

Voice over IP (VOIP) has spread quickly within corporate environments to replace ageing PBX exchanges. VOIP provides the digitisation of voice traffic and is normally spread throughout an organisation using the same network switching fabric that is used for regular network traffic, albeit it is normally split off onto a separate virtual network (VLAN). VOIP opens the doors for soft-phones that allow software on an employee’s laptop to receive phone calls and use the built in microphone and speakers of the laptop. Desk-based VOIP phones are also popular as they provide a mechanism to allow hot-desking and add a layer of security by allowing calls to be tracked and controlling the types of call that can be made.

In most instances, the VOIP boundary terminates at the telephone exchange where it meets the copper lines used by conventional telephone systems. However, organisations are increasingly allowing direct VOIP calls out of their network infrastructures. This opens another channel for an attacker who wishes to communicate with their APT. By encoding the data they wish to send into audible codes, the attacker may be able to create a communications channel using VOIP systems. This attack was implemented [27] recently and has been used to highlight the dangers of poorly configured VOIP infrastructures.

The potential use of VOIP as a C2 channel is often be overlooked as most implementers and auditors are concerned with its capabilities as a listening device, but it can be used to add modem-type capability to an APT infected machine on a VOIP-enable network. An
APT may be able to use the VOIP system to dial out to a C2 server, using SIP or even a traditional phone call, to pull commands and to exfiltrate data. It may also be possible for the attacker to contact the APT directly if a direct-dial VOIP address or telephone number can be determined.

A.7.2 Email

Even in the most heavily secured networks, email is often allowed inbound and outbound from the Internet. Email filtering appliances and services have advanced through the years to protect users from spam, viruses, and malicious content. However, seemingly non-malicious content may still be entering and exiting a network which, whilst looking benign, can contain commands for a hidden APT. Email provides a simple mechanism of directing messages to a specific user within an organisation and can be used to target specific APT infections.

To form a C2 channel using email, the APT will need to be hooked into the email system in order to monitor incoming email and to be able to send emails. This may be achieved in two ways: either by directly connecting to the mail servers or by process manipulation of a mail client installed on an infected host. The APT will register itself with its master by sending an email from the source mailbox to a known or calculated address, alerting the attacker that a new infection has occurred or resumed. Email messages directed to that source mailbox will contain commands to be executed, and email will be used to return results. These steps can be used to create a Push model C2, and map directly onto the C2 channel communications model defined earlier.

The chances of being discovered are increased if the APT uses an active mailbox for receiving commands and sending results - the user may happen upon large volumes of sent mail and become suspicious. Furthermore, the attacker will need to be careful that the volume of email generated is in line with that of a regular user and sent within the working day. There have been studies [27] into the applicability of using email as the C2 channel through the use of email attachments and steganography to hide messages destined for an APT. Similar obfuscation and timing considerations will be common across all types of C2 channel.

A.7.3 Further threats

As existing protocols and methods are detected and increasingly blocked by organisations, ever more novel methods will be used. This has been demonstrated during the writing of this report when researchers at the Fraunhofer Institute have demonstrated a method of communicating between computers using inaudible audio signals rather than conventional network connections [28]. While the current bandwidth is low (approximately 20 bits per second) this could still be used as part of a C2 channel to infected machines that are not connected to a publicly accessible network.

While such attacks are likely to be reserved for high-value targets, it is a prime example of how the defence against C2 channels is a never-ending arms race. An organisation cannot be complacent that it is safe because it is defending against all known C2 channels, as future developments will bring new and unexpected channels that are not being defended against.
Annex B C2 and the Council on CyberSecurity’s Top 20 Controls

The Council on CyberSecurity is an independent, expert, not-for-profit organization with a global scope that provides analysis and advice on best security practice to cyber security professionals the public. As part of this work they have taken stewardship of twenty “Critical Controls for Effective Cyber Defense” [30] from the SANS Institute. This was previously also presented by the SANS Institute as an easily accessible poster [31].

The controls provide a baseline level of security that all organisations should implement to secure their networks, and prioritises some “quick wins” for those networks that are currently undefended. The list was colloquially known as the “SANS Top 20”. The full document provides more detail on each control, including metrics to assess success. The list of named controls in version 4.1 of the Top 20 is:

- Critical Control 1: Inventory of Authorized and Unauthorized Devices
- Critical Control 2: Inventory of Authorized and Unauthorized Software
- Critical Control 3: Secure Configurations for Hardware and Software on Mobile Devices, Laptops, Workstations, and Servers
- Critical Control 4: Continuous Vulnerability Assessment and Remediation
- Critical Control 5: Malware Defenses
- Critical Control 6: Application Software Security
- Critical Control 7: Wireless Device Control
- Critical Control 8: Data Recovery Capability
- Critical Control 9: Security Skills Assessment and Appropriate Training to Fill Gaps
- Critical Control 10: Secure Configurations for Network Devices such as Firewalls, Routers, and Switches
- Critical Control 11: Limitation and Control of Network Ports, Protocols, and Services
- Critical Control 12: Controlled Use of Administrative Privileges
- Critical Control 13: Boundary Defense
- Critical Control 14: Maintenance, Monitoring, and Analysis of Audit Logs
- Critical Control 15: Controlled Access Based on the Need to Know
- Critical Control 16: Account Monitoring and Control
- Critical Control 17: Data Loss Prevention
- Critical Control 18: Incident Response and Management
- Critical Control 19: Secure Network Engineering
- Critical Control 20: Penetration Tests and Red Team Exercises

B.1 The Critical Security Controls and C2

All of the Top 20 Controls are important for maintaining a secure network, however only a subset of the controls are relevant to combating C2 channels. With reference to the list above, the following controls can be said to directly aid in the defence against C2 channels. Where an emphasis is placed on specific sub-controls then this is not an indication that these should be done at the expense of other controls for general security, but that these are particularly relevant to the C2 aspects of the security threat.

Controls 2 and 3 describe the process of locking down the software and configuration on your machines and identifying misconfigurations. While some malware might compromise a legitimate service or application and not have a distinct footprint, some malware will exist as their own executables that could be detected or prevented through these methods. The focus for combating malware using C2 channels should be in the sub-controls that are described as “Configuration/Hygiene”, as these will identify deviations from the norm that can be caused by malware, especially those running C2 channels.

Control 14 describes the process of monitoring a system to compare it to expected behaviour. Monitoring is an important part of trying to detect C2 channels. Sub-controls 14.4 to 14.9 and 14.10, in conjunction with the techniques described in the Defence by Monitoring section of the main body of the report, will provide opportunities to identify unusual communications or behaviour patterns caused by C2. Sub-control 14.11 will help to detect the spread of an infection, which could be directed by commands from a C2 channel.

Controls 4 and 20 suggest a process of pro-actively checking the security of a network. With appropriate penetration testing, a network can be checked to see how advanced a C2 technique needs to be to be able to establish itself. Any remedial action taken as a consequence of the penetration test would then improve the defences of the network against C2 channels. Sub-controls 4.2 and 4.4 will support the monitoring process against C2 channels, while sub-control 4.9 may identify processes used for C2. Sub-control 20.1 summarises the need to perform penetration tests on a network, which should include attempts to compromise a network by using C2 channels. This is only useful if sub-control 20.4 is then completed, mitigating any compromises that were identified.
Control 5 describes standard defences against malware and exploitation. In combination with additional application behaviour whitelisting (see later) this can prevent some compromises and, with the right signatures, may be able to detect the C2 channels of other compromises. Sub-control 5.9 and 5.17 will block or disrupt a subset of malware, while sub-controls 5.1, 5.2, 5.10, 5.12 and 5.14 to 5.16 will allow the detection of some channels.

Controls 7, 10, 11, 13 and 19 relate to securely designing and implementing a network. Due to the variety of methods that could be used for each C2 channel then the only network that could be guaranteed not to have C2 channels is a stand-alone network. However, with appropriate security controls, breaks and filtering in a well-designed network then a proportion of C2 channels can be prevented or disrupted. All of the sub-controls will contribute to this and reduce the available bearer methods for C2 channels, but sub-controls 10.2, 10.3, 11.2 and 13.6 capture the simplest and most effective way of disrupting a subset of C2 channels. Sub-controls 19.1, 19.3 and 19.4 provide more advanced disruption of C2 channels through network architecture decisions.

Sub-control 13.1 is a similar quick-win, but its relevance is changing due to the increasing prevalence of peer-to-peer and pop-up C2 command networks. All of the other sub-controls of control 13 are also relevant to blocking or detecting C2 channels, with the exception of sub-controls 13.3, 13.7 and 13.8.

Control 9 covers training staff. Social engineering – where the human in the chain is encourage to perform an action that leads to compromise – is an increasing problem as security measures improve. Well trained staff are less likely to perform actions that might cause an infection that triggers a C2 channel, while well trained network operations staff will be better able to detect, understand and defend against C2 attacks. Most of the sub-controls relate to general malware protection, but sub-controls 9.2 and 9.3 are most relevant to the on-going detection and disruption of C2 channels on a network through the training of the network’s management and monitoring staff.

The corollary of this is that the following controls are not relevant to combating C2:

Control 1 is irrelevant as the use of C2 almost certainly means that the attacker outside of the corporate boundary and so no hardware will be added to the network and so the existence of an inventory wouldn’t help in detecting the changes that compromises with C2 channel cause. The control is indirectly relevant, however, in terms of understanding what your network looks like and which machines should be running which services and communicating with which other machines. This understanding will help to spot anomalous traffic.

Control 6 is relevant to malware in general, but does not have any C2-specific aspects as it focuses on how to design and build secure software and systems.

Controls 12, 15 and 16 relate to user accounts and permissions. This will have limited effect on the C2 channel itself, but minimal permissions, enforcement of “need to know”, and disabling unused accounts should reduce the likelihood of initial compromise and reduce the impact of the activities undertaken at the direction of the C2 channel.

Controls 8, 17 and 18 cover disaster recovery and data loss. While some aspects of this are relevant to clean-up after a C2 incident, they are not C2 specific and will not prevent or limit a C2 channel in and of themselves.

### B.2 Additional Controls

While the Top 20 Controls list provides a good basis for all system administrators to secure their network, high value and high risk networks will need additional security measures to further secure them against more advanced attack. In addition to the Top 20 Controls, the following measures and extensions may help reduce C2 incidents. The value of each control will depend on the value of and risk to the network in question.

**Application behaviour whitelisting:** Tighter whitelisting of applications in the form of mandatory access control on applications (such as AppArmour and SELinux on Linux) may further improve security beyond that provided by Controls 2 and 3 by preventing compromised executables from accessing unexpected files, network connectivity or resources that could be used to establish C2 channels or as part of the activities undertaken under the command of a C2 channel.

**Split DNS:** Where proxy servers are used for all outbound connections from an intranet (as described in Control 13) then that network need only resolve internal addresses, while all external addresses are resolved by the proxy server. By appropriately segregating the responsibility for DNS traffic so that internal DNS servers never look up external name servers then it is possible to prevent direct DNS tunnels and monitor the activity. This is an extension of Control 19.3, which still allows a C2 channel to use DNS, albeit...
in a method that would be readily logged and more easily detected than if the control was not in place.

SSL breaks: While HTTP traffic can be configured to use a proxy server as part of Control 13, so that the proxy server can monitor and filter the traffic for C2 channels, HTTPS traffic is normally allowed to pass through the proxy server as-is because it is encrypted. This obviously limits the ability of a monitoring system to determine whether data is being exfiltrated over HTTPS. For particularly high risk and high value network where the system administrator has full control of the end nodes then an SSL scanning appliance can be used that breaks the HTTPS connection into an Internet to appliance link (encrypted with the website’s certificate) and an appliance to end node link (encrypted with the appliance’s certificate, which is configured as a trusted certificate on the end node). This will allow HTTPS traffic to be scanned, but consideration needs to be made to ensure that users are appropriately aware of this activity due to its nature as an authorised Man-in-the-Middle compromise on their HTTPS connection and that the activity is undertaken in line with local legislation and policy.

Remote Forensic Agents: While malware behaviour and C2 channels can often be spotted by their network activity, it may also be possible to spot them post-infection but prior to network connectivity by monitoring the behaviour of processes on the nodes. Such monitoring generates large amounts of data, but gives a detailed picture of which processes and applications are performing which activities on a machine.

Monitoring of traffic for protocol compliance: This extension to Control 11 may prevent some C2 channels by preventing, for example, a HTTP proxy from being used as a tunnel for C2 traffic using the IRC protocol.

Monitoring of traffic for later analysis: As an extension to Control 14, a full packet capture on the network can help to further locate information about compromises. As with the existing control, this will help future analysis of the incident and improve the knowledge of the network administrators. This is also effectively an extension of sub-control 13.2 that brings the monitoring from the perimeter and the DMZ into the main network to detect the starts of a C2 and initial malware activity before it attempts to connect out of the network.

Behaviour modelling: The monitored network must be understood to support the later analysis of captured data. While this knowledge can be captured at the level of which machines are servers and which are desktop machines, a process of more complete behaviour modelling can help to further identify anomalies. Although network engineers may be expected to perform DNS queries or trigger ICMP packets as part of their duties, a normal office worker is unlikely to perform these activities and so such activity could be indicative of an attempt to initiate a C2 channel. This level of anomaly detection is not available with a simple desktop/server split.

Correlation of data sets: Additional data sets should be used, where possible, to further supplement the analysis of captured data. Although some traffic patterns may not appear particularly unusual on their own, if the behaviour is occurring on a day that the HR system shows the employee is off work on leave or sick then further anomalies can be detected. This can help to detect C2 channels that are indiscriminately connecting to the network while being unaware of a user’s real behaviour. Similar inferences could be made with other logs, such as building access logs being compared to usage patterns and machine locations to identify operations performed under a user’s username when they are known to be on another site.

Big Data analysis of monitoring: The monitoring described above and in Control 14 can generate large volumes of data. The best analysis results are achieved by storing all of this data for long periods of time and looking for deviations from long-term patterns and norms. With moderate to large networks, or particularly busy networks, this may require Big Data techniques to support ad-hoc processing over large volumes of data for the extraction of patterns that are indicative of either malware infection or, more specifically, potential C2 channels.
QinetiQ. Cyber security you can trust.
For more information on how we can help you, please contact us on +44 (0)843 658 4668