Security Lancaster | LANCASTER UNIVERSITY

Detecting and Preventing Data Exfiltration

Contributors:

Professor Awais Rashid,
Dr. Rajiv Ramdhany,
Matthew Edwards,
Sarah Mukisa Kibirige,
Dr. Ali Babar,
Professor David Hutchison,
Security Lancaster,
Lancaster University, UK

Dr. Ruzanna Chitchyan,
University of Leicester, UK

GCHQ Academic Centre of Excellence in Cyber Security Research

EPSRC
Pioneering research and skills

Introduction

Circa 2014. The cyber security landscape or the 'ThreatScape' as we call it, has changed, again. The emergence of new business models and technologies has seen a recent radical increase in vulnerabilities to which organisations are exposed. Modern organisations now operate as part of a complex, partially trusted eco-system comprising other organisations, a diverse range of third-party technologies, and end-users operating in a variety of organisational cultures. The need to meet organisational and business challenges such as increased productivity, lowered costs, and customer satisfaction has driven organisations to break the traditional 'silo' infrastructure model as they try to improve efficiency, engage more with customers and collaborate closely with partner organisations. Organisations have embraced cultures where employees work anytime and from anywhere.

They have adopted the use of informatics and social business models for closer collaboration with business partners and with their customer base. The need to engage with customers and other service-providers globally has led to the rapid adoption of cloud-based services and virtualisation. As organisations grow in size and complexity, they are increasingly harnessing the power of Big Data analytics to leverage the value from the tremendous volume of data they generate. Systems are instrumented to gather more contextual data from transactions, move data across locations, and store it for the purposes of data-mining and sophisticated data analytics. Data (whether it is personalised information, intellectual property or trade secrets) counts as an organisation's most valuable asset.

For organisations to rise to these challenges, their IT infrastructures have experienced significant transformations in terms of heterogeneity and complexity. As a result of this, the cyber-security model where perimeter defences maintain a bastion of control over an organisation's network (and crucially, its data assets), no longer suffices for protection. Opening up the infrastructure through internet- and web-based interfaces to support employee mobility and interactions with other organisations (as part of a supply chain, Software as a Service) has resulted in more entry points that can be attacked by cyber criminals. 'Bring your own device' cultures whereby end users utilise new personal technologies or software services (from partially-trusted third parties) in their day-to-day working practice have introduced new security challenges and the threat of additional attack vectors. IT infrastructure use is no longer restricted to an organisation's employees but extended to customers, contractors and outsourcers, thereby creating the opportunity for new attack threats. In essence, an organisation's IT infrastructure is often a patchwork of systems and technologies procured from third-party providers which can be harbingers of latent vulnerabilities. As system procurers have little control on the design of and the security controls built into these sub-systems, they can only be partially trusted at best. Such partially-trusted settings are not an exception but have become the norm in modern business settings.

CPNI
Centre for the Protection of National Infrastructure

Security Lancaster and the University of Leicester would like to acknowledge the help and support of the Centre for Protection of National Infrastructure (CPNI) in producing this document and associated resources.

Social Business
Cloud and Virtualisation
Employees
Customers
Contractors
Outsourcers
Bring your own IT
BRING YOUR OWN
Mobility, SaaS

There has been an evolution on the cyber-crime front as well where attack campaigns have seen a shift in motives and an increase in sophistication. Due to their reduced impact, insiders, spammers, worm and virus writers are now considered more of a nuisance than a threat. Hacktivitism is seen to have more notoriety-seeking and defamation motives as activists such as Lulzsec and Anonymous undertake campaigns to cause systems disruption, web defacement and information disclosure. Although hacktivists remain a nuisance, the most potent threats seem to now stem from organised-crime, industrial espionage from competitors and cyber espionage by nation states. A significant number of data breach incidents[1;2] (Zeus, ZeroAccess, Blackhole Exploit based attacks) point to organised criminal gangs as a new breed of cybercrime perpetrators who actively seek to steal data from organisations for monetary gains. Advanced Persistent Threats (APTs) and nation-state actors also list among this new breed of sophisticated perpetrators, whether it is for the theft of intellectual property for financial/competitive gain or targeted attacks on critical infrastructure for strategic and economic motives. The Stuxnet, Flame, Aurora and Duqu attack campaigns are examples of the virulence of the new threats and the level of sophistication seen from this group of attackers.

Attack tools have evolved as well. Botnets, backdoors and malware are often weapons of choice and served up by Phishing attacks. Malware and hacking tools are now written by professional programmers, often by contract and designed to avoid detection by intrusion detection systems. Exploit kits that include functionality such as rootkits, keyloggers, backdoor access as well as bot herders are available either for free or for purchase by cyber criminals.

Sophisticated attackers, the availability of sophisticated hacking tools and the radical increase in system vulnerabilities are the primary concerns in this new security landscape. In fact, the perimeter defence model is now dead. As the opportunity for attackers to gain a foothold in an organisation's network has risen dramatically, intrusion detection/prevention strategies are now deemed to be inadequate for data protection. Data exfiltration, whereby an attacker having established presence on an organisation's network can filter data out of the system through a variety of means, is a very potent threat. The nature of the threat is such that as well as erecting the traditional 'castle defences' (IDS/IPS), organisations must focus on protecting their 'crown jewels'. Defences today need to be data-centric and embody strategies to detect, prevent and mitigate data exfiltration attacks.

This document is designed to help you understand how to develop a robust approach to detecting and preventing data exfiltration from your organisation. It covers the various modalities in which data exists within an organisation and how remote attackers exploit readily-available means of data exchange prevalent in most organisations as well as more advanced modes that are aimed at bypassing most security measures. The key message here is that the contents of an attacker's arsenal are diverse and sophisticated attackers such as Advanced Persistent Threats (APTs) are not limited to using typical channels to extract data from organisational networks. They have the capability, motivation and resources to use more advanced channels.

Security measures in an organisation, therefore, need to account for these various data exfiltration modes. In this regard, the document reviews both overt and covert exfiltration channels and describes how data exfiltration along these various channels may be detected and prevented. Case studies are included to demonstrate how to implement these measures in concrete settings. An extensive list of scientific articles on modes for data exfiltration and its detection and prevention is provided. The report concludes with an analysis of emerging technological trends, the various opportunities they provide to remote attackers for data exfiltration and guidelines on how an organisation's security posture may be hardened against these emerging threats.

Extensive web-based resources including an animated info graphic that enables users to explore various data exfiltration modes and counter measures, additional case studies, and links for further reading complement the report.

Where are your Data Assets?

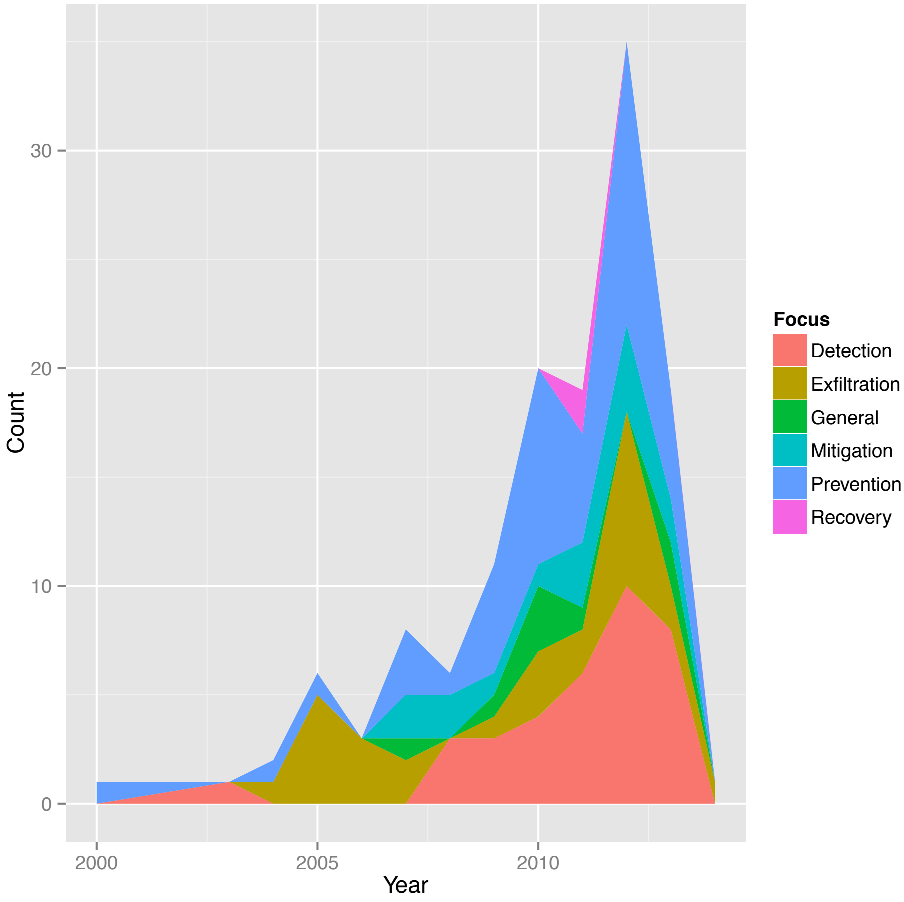
In the modern digital society, 'data' is one of the biggest and most valuable commodities. It is, therefore, a key motivator for a number of cyber attacks, be it to gain access to personal data, financial information or intellectual property. At the same time, the complexity arising from modern business settings, new technologies and resulting modes of work lead to an increase in the cyber-attack surface of an organisation and, consequently, a higher risk of data exfiltration. Advanced Persistent Threats (APTs) and attackers motivated by industrial espionage are persistent and return again and again in order to achieve their objective[3].

The costs of such data exfiltration are substantial – with fines from regulatory bodies and cost of IP theft leading to significant economic losses. The industry report produced by PWC and published by Info Security Europe on information security breaches[4] provides the current industry benchmark for the state of play with regards to the cyber security issues and experiences of the business community. The report identifies a significant increase in the attacks on small to medium businesses and an almost doubling of the cost of the worst incident as a result of those attacks. The latest Cyber Security Intelligence Index from IBM[5] estimates an average of 73,400 cyber security attacks and 90 incidents (where some sort of breach has occurred) in a single organisation per year. It also estimates that 50-75% of these attacks arise from malicious external actors and 23% are motivated by industrial espionage, financial crime, terrorism or data theft.

If we accept that there is no perfect cyber security, then the principle of 'resilience' becomes a key driver for any security measures:

If we assume that at some point a breach in the security measures will take place, what mechanisms can enable cyber security personnel and systems in an organisation to detect, stop or at least disrupt data leakage from the organisation's system following such a breach?

Focus of paper by publication year



This graph shows the results of an extensive literature review based on key terms linked to data exfiltration.

Four key phrases were searched for in seven computer science digital libraries, with no restriction on date of publication. This resulted in 3,142 results. These results were then manually filtered for relevance and quality via two rounds of screening. Where a paper was judged irrelevant or of low quality by a reviewer, it was discarded. This procedure resulted in 132 papers making it through to inclusion in the final dataset.

The papers were then assigned labels indicating their primary focus with regards to data exfiltration, along with the technical level of their output and the methodology behind the paper.

The graph shows the primary focus of papers as displayed over the years included in the publication dataset. We can, for example, observe that papers discussing Mitigation strategies are first seen in 2006, and papers discussing Recovery strategies are first seen in 2010. We can also note that Recovery has received little attention to date.

To do so, one needs to understand the various modalities in which data exists within an organisation and put in place counter measures that can detect and prevent typical data exfiltration tactics employed by attackers, mitigate the threats posed by more advanced tactics and put in place recovery strategies in case attackers manage to circumvent security measures. The latter is a key to ensuring that security measures are not blind to the constantly changing threat landscape and the increasingly sophisticated tools and tactics that are at the disposal of mal-actors such as Advanced Persistent Threats (APTs). However, our analysis of data exfiltration literature since the year 2000 shows that recovery from data exfiltration has received little attention to date with most approaches focused on detection, prevention and mitigation.

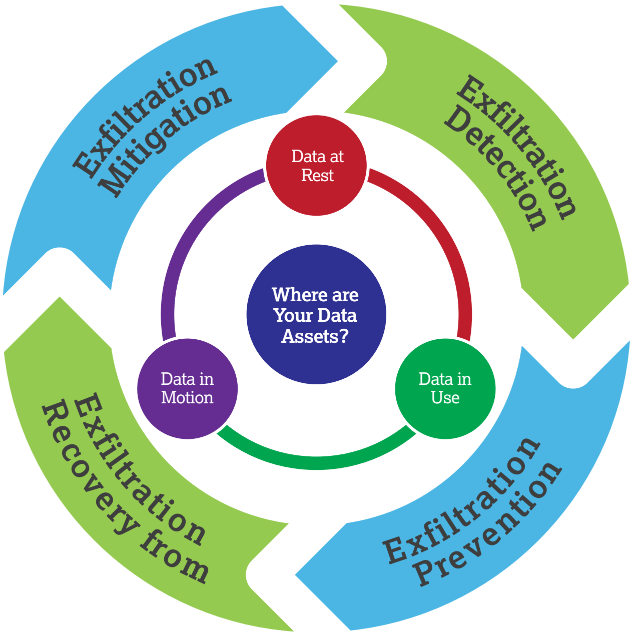
Data at Rest

Data at Rest refers to 'inactive data' stored physically in databases, data warehouses, spreadsheets, archives, tapes, offsite backups or on mobile devices. It includes but is not limited to archived data, data which is not accessed or changed frequently, files stored on hard drives, and files stored on USB thumb drives, backup tape, disks and offsite storage. Data at Rest may be subject to occasional changes although not with the same frequency as Data in Use ('active data' held in a database or being manipulated by an application). Because of its worth as an organisation's asset, the security of Data at Rest is an increasing concern to businesses, government agencies and other institutions.

With the emergence of 'bring your own device' culture, mobile devices are often subject to specific security protocols to protect Data at Rest from unauthorised access when lost or stolen and there is an increasing recognition that database management systems and file servers should also be considered as at risk. The longer data is left unused in storage, the more likely it is that unauthorised individuals outside the network might retrieve it.

Data in Use

Data in Use refers to 'active data' assets under constant change as they are processed by applications. They are usually held in non-persistent storage such as computer memory, CPU caches and CPU registers. Data in operational tables in databases is also classified as Data in Use as opposed to data in reference tables. The latter are updated infrequently and known as Data at Rest.



Data in Use, or in memory, can contain sensitive information such as digital certificates, encryption keys, intellectual property (software algorithms, design data), and personally identifiable information. Compromising Data in Use often enables access to encrypted Data at Rest and Data in Motion. For instance, an attacker with access to random access memory can parse that memory to locate the encryption key for Data at Rest and decrypt it. Thus, because of its nature, the security of Data in Use is of increasing concern to businesses, government agencies and other institutions.

Threats to Data in Use can come in the form of cold boot attacks, malicious hardware devices, rootkits and bootkits.

Data in Motion

Data in Motion is data that is traversing the enterprise network, or temporarily residing in computer memory to be read, updated or forwarded to another data-processing service. Within an organisation, data is in constant movement, from the moment it is captured, and manipulated by the algorithms that encapsulate the business logic of the organisation to when it is committed for storage on disk or in databases. When data or files are in transit whether in the form of packets transported by communication protocols or as data held in computer memory or printer memory buffers, there is ample opportunity for an unauthorised person to intercept and extract data.

As users access applications and the sensitive information that they expose, data exfiltration countermeasures are needed to thwart leakage threats as the data moves between workstations, servers and host-based applications. Likewise, these countermeasures must be integrated into the existing network infrastructure to provide protective, not just detective controls when files are transferred from users to users, from users to servers, and from servers to servers within and outside the organisation.

Anatomy of Typical Data Exfiltration Attacks

Typical data exfiltration scenarios involve the attacker establishing a Command and Control (C2) connection to remotely control the compromised machine. This entails the attacker infiltrating the host computer by exploiting security vulnerabilities (including the human element) in the organisation. For instance, a phishing email such as one containing weaponised Word documents (CVE-2012-0158), Adobe Acrobat PDFs (CVE-2009-4324) and Microsoft Help Files (.HLP) could be used to entice an employee to infect the host computer with a virus (that installs a Remote Access Tool (RAT)) or with a backdoor. A backdoor is a piece of malicious code deployed on a compromised computer to facilitate C2 and exfiltration. Remote Access Tools and backdoors are types of malware that use the C2 set up to exfiltrate data out of organisations. They are weapons of choice in the arsenal of tools designed to perform targeted attacks by Advanced Persistent Threats (APTs). C2 are often HTTP based. Commands are received over what looks like a regular web browsing session to analysts inspecting traffic. The attacker's end point is an attacker-owned IP address or domain. Many variations are possible (for example, IRC, FTP, Peer to Peer networks, watering hole, etc.) for exfiltration of data.

Data Exfiltration Strategies

When it comes to data exfiltration, the simple approach seems to be the most effective as a large number of organisations are not set up to effectively counter exfiltration attempts. Security measures are often focused on perimeter protection and do not start from the key assumption that persistent attackers would gain access and how to detect and disrupt their activities, especially their attempts to compromise data assets, once they have established a presence. Exfiltration via outbound FTP or HTTP/HTTPS connections is most common (more than 50% of data breach incidents we have analysed favour these exfiltration modes). It blends in with normal network traffic and is hard to distinguish from legitimate activities of users. Attackers also use a variety of data exfiltration strategies ranging from an indiscriminate file dump that takes the data offline for later analysis/processing to very careful and considered filtering of the data to extract only the most pertinent and high-value information.

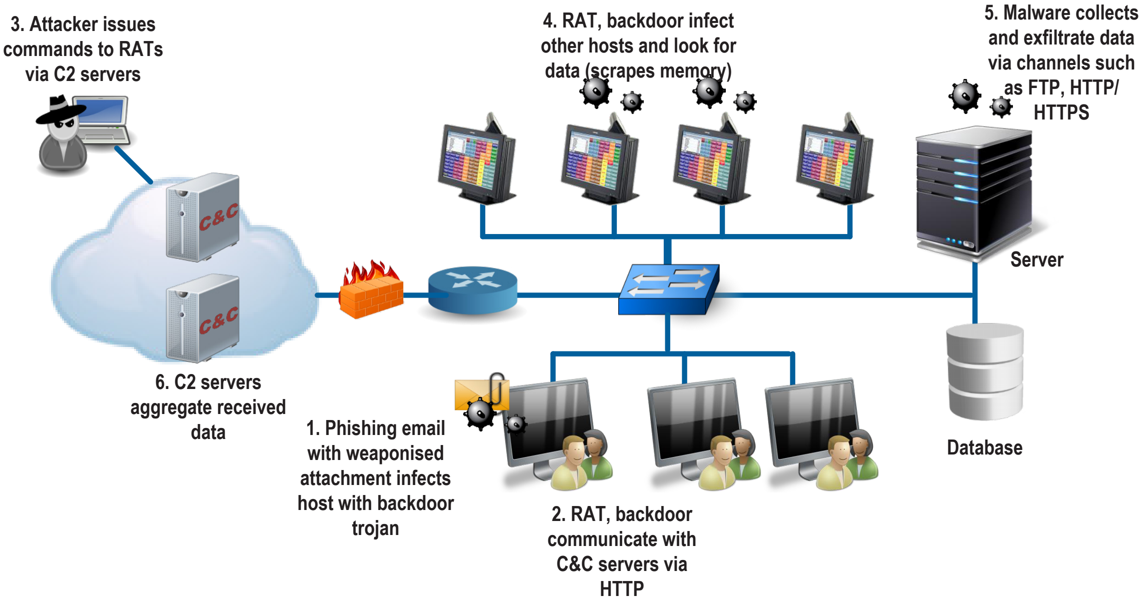
Attackers also think of Resilience!

Sophisticated attackers such as APTs take into consideration the fact that the connectivity between the payload (RAT, backdoor) and the attacker cannot be assured for long periods of time (and sometimes is consistently non-existent). As such, the C2 schemes used by such attackers account for this and ensure that the payload is well equipped to operate fairly independently.

Nevertheless, some form of control communication is needed, and usually utilises a hierarchical control structure, where multiple payloads are deployed into the organisation at different locations, and are able to communicate with each other to form a 'grid' that enables the more restricted locations to communicate through other layers to the attacker outside the organisation.

In-band vs out-of-band exfiltration

In-band exfiltration utilises the same channels as C2 to extract data. However, sophisticated attackers have the capability and will use out-of-band exfiltration where the exfiltration channels and end-points are distinct from C2 channels and end-points. In out-of-band exfiltration attackers maintain separate external drop points ensuring C2 resilience if the data exfiltration channel is detected.


3. Attacker issues commands to RATs via C2 servers
4. RAT, backdoor infect other hosts and look for data (scrapes memory)
5. Malware collects and exfiltrate data via channels such as FTP, HTTP/ HTTPS
C&C
C&C
Server
6. C2 servers aggregate received data
1. Phishing email with weaponised attachment infects host with backdoor trojan
2. RAT, backdoor communicate with C&C servers via HTTP
Database

Partially-Trusted is the Best Philosophy

The deployment of perimeter firewalls, proxy servers and access control lists can help block access to C2 servers. Attacker-owned IP addresses or domains can be blocked or the domains blacklisted in DNS sinkholes. An increasing number of C2 schemes are, therefore, leveraging the use of trusted sites for data exfiltration. This is because these sites are, well, trusted and thus data exfiltration traffic that flows towards them would easily blend in with legitimate network traffic and not appear suspicious to Data Loss Prevention (DLP) systems. Trusted sites would also not be included in threat intelligence feeds that list known bad domains or IP addresses. They also do not show up in Least Frequency of Occurrence reports where outlier analysis is used to detect anomalous data transfers to particular domains or IP addresses. During exfiltration forensic investigations, often traffic to these sites is not scrutinised due to sheer volume.

Thus, malware implants on compromised workstations can use known file-sharing services like Box.net or DropBox to take data out of organisations. The implant also receives C2 instructions over the same channel. The attraction to the attacker is that a single DropBox account can be used as a C2 control point and as a collection point for extracted files; the attacker can access this account from anywhere. The file-sharing service account is used to upload files, download malware updates, upload information about target hosts such as process listings, running services and drive list, execute arbitrary commands and upload results, and configure malware sleep intervals.

In addition to acting as a C2 endpoint, trusted sites are increasingly being used by more sophisticated attackers for locating C2 servers. The contemporary approach by malware developers has been the use of DNS instead of direct IP addresses for communication with C2 servers; this renders their malware more survivable to changes of the C2 servers' host IP address. However, 'bad' domains are increasingly likely to end up on blacklists; DNS sinkholes are becoming more commonplace as a countermeasure for detecting and blocking malicious traffic, and to combat bots and other unwanted traffic. In this respect, malware developers have been looking at alternative non-conventional ways to implement DNS-like systems and still achieve the survivability benefit of not being tied to a single static IP address. So, the attackers instead of resolving a domain name to the IP address of a C2 node, distribute the IP address on a trusted location such as a social media site. One particular malware encountered in the field, when analysed, was found to use Twitter as well as an obscure Guest Book site (as a fail-over) for IP resolution of C2 servers. In this particular case, the IP address of the C2 node was provided in encrypted form on a Twitter profile.

Case Study: Targeted Data Exfiltration by Advanced Persistent Threats (APTs)

APTs are complex (i.e. advanced) cyber attacks against specific targets over long periods of time (i.e persistent). Originally, the term was used to describe cyber-espionage by nation states stealing data or causing damage to critical infrastructure of other nation states for strategic gain. However, recent targeted attack campaigns have seen APTs steal data from businesses for financial or competitive gain. A recent data breach at RSA was one of many targeted attacks by APTs. In contrast to other types of data exfiltration attacks, APT attacks are *targeted, persistent, evasive and complex.*

Targeted: In contrast to the more random opportunistic attacks against a large number of targets, APTs target specific organisations with the purpose of stealing specific data or high-value intellectual property. Operation Aurora, the APT that targeted a number of high-profile companies like Google and Adobe, was motivated by access to source code and possibly geo-political factors such as access to email accounts of Chinese dissidents and human-rights activists. The Sony Playstation Network attack by LulzSec targeted personal user information such as users' names, addresses, birth dates, email addresses, passwords, logins, handles, profile data, purchase/billing history, and password security answers. The RSA APT attack targeted intellectual property such as SecurID keys that are necessary to access many encrypted systems used by the U.S. government, intelligence agencies, defence contractors, and Fortune 100 companies. In essence, APTs are not opportunistic attacks targeting just about any organisation with a vulnerability to a known exploit. Rather, they are focused campaigns perpetrated by individuals with the time and resources to achieve their objectives.

Persistent: APT attacks develop over a long period of time, usually in multiple phases. To steal data, the attackers must identify vulnerabilities that exist in the organisation's infrastructure, evaluate existing security controls, gain access to privileged hosts within the network, find valuable data, and exfiltrate the data via suitable channels without raising suspicion. Discovering where valuable data resides in an organisation, detecting what security controls are in place and what vulnerabilities exist that might be exploited take time. The entire process may span months or even years. Due to its persistent nature, the detection of APT attacks cannot, therefore, rely on any single event, but should instead look for patterns of events over time, that are characteristic of APT methodologies.

Evasive: In order to avoid detection and maintain a clandestine presence to maximise the chances of locating and extracting valuable data, evasive manoeuvres are taken by APTs to bypass the traditional security solutions deployed in most organisations. For instance, to gain access to hosts on the target network whilst avoiding perimeter firewalls, the attacker may deliver threats (for example, an Adobe Flash Zero-Day exploit) within content carried over commonly allowed protocols (e.g. smtp, http, https, etc). Further to avoid detection of malware by antivirus software, attackers often write custom code (or modify existing malware) to suit the environment. As these malware variants have not been seen prior to the attack, antivirus signatures are not available to provide protection. As another evasive mechanism, APTs use custom encryption and tunnel contents within protocols that are allowed through the firewall.

Complex and Sophisticated: When organisations are guilty of system mis-configuration or poor software design, even a crude APT is sufficient to exploit system vulnerabilities and steal data. However, APT-actors can resort to sophisticated tactics to achieve their goal. APTs are complex; they use a mix of different attack vectors and schemes targeting multiple vulnerabilities identified within the organisation. An investigation into the Linked-in profile of an organisation's employees may reveal the technologies used and point to exploits that may be applied to take advantage of particular vulnerabilities (for example, buffer overflow exploit of a specific web server). Social engineering tactics such as spear-phishing may then be used to target key individuals in the organisation. Phishing emails could be sent to the key individuals with links to a website that executes custom JavaScript code to install a remote access tool. Many different types of malware may be downloaded to establish a C2 structure for exfiltration. Multiple-levels of encryption may be applied to malicious code to avoid detection. Due to the diversity of attack vectors, no single security control provides protection against all of these vectors. Multiple detection mechanisms must be used together in a multi-layered approach to identify complex patterns of evasive behaviour.

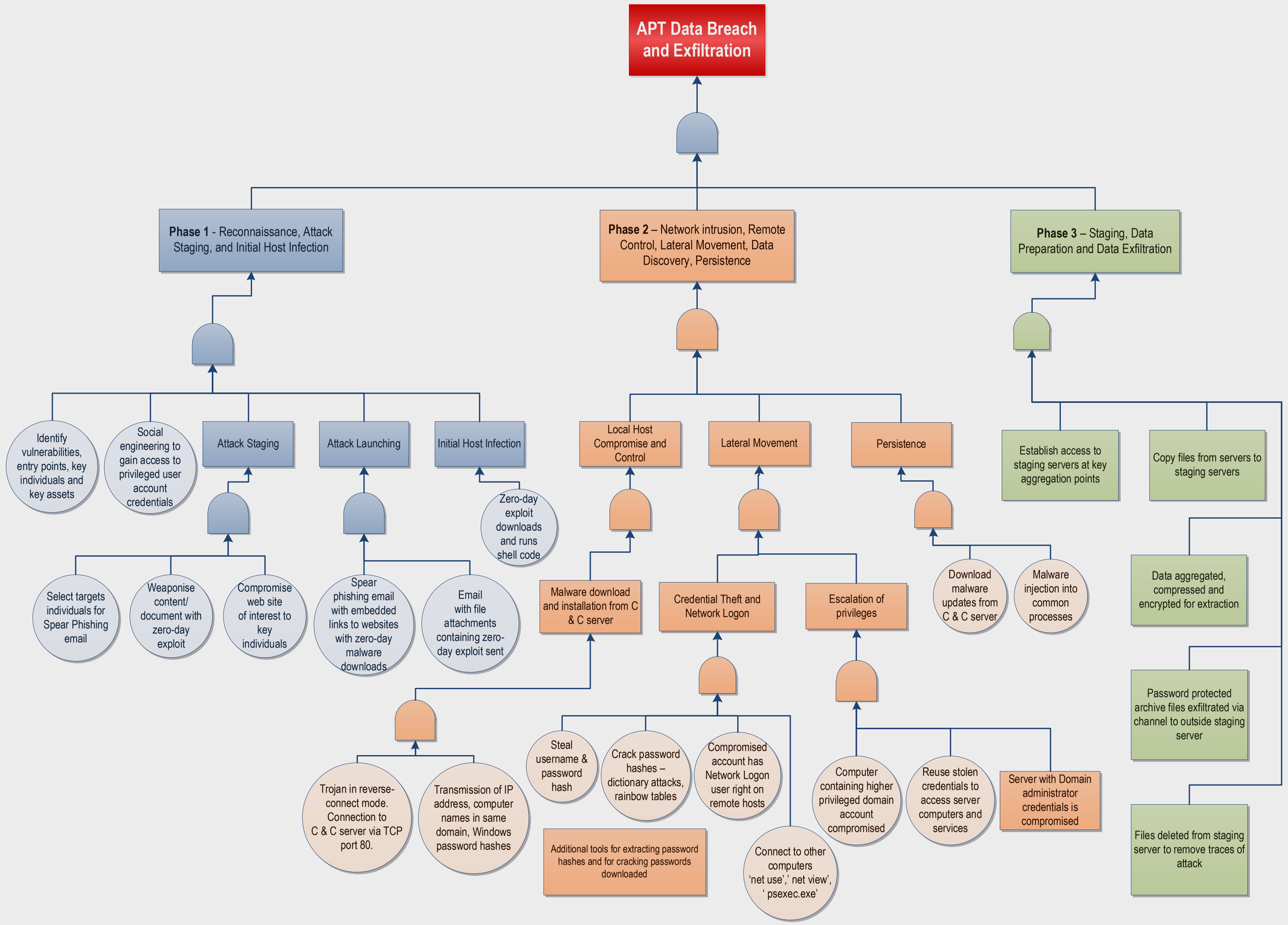
Anatomy of Data Exfiltration Methodologies of APTs

Our analysis of multiple APT case studies such as Operation Aurora and the RSA data breach has revealed that an APT attack process spanning a period of months has three distinct phases.

Phase 1 - Reconnaissance, Attack Staging, and Initial Host Infection: The attacker performs reconnaissance, identifies vulnerabilities, decides on attack vectors and launches the attack to infect the first hosts.

Phase 2 – Network intrusion, Remote Control, Lateral Movement, Data Discovery, Persistence: The attacker pursues the initial network intrusion to establish remote control of infected host machines, download code, spread the malware infection to other machines, and move laterally and escalate privileges in the search for data repositories.

Phase 3 – Staging-Server Selection, Data Preparation and Data Exfiltration: The attacker establishes data aggregation points in the network to collect data, copies the data to these endpoints, prepares it for exfiltration and, finally, extracts the data from the target network.


APT Data Breach and Exfiltration
Phase 1 - Reconnaissance, Attack Staging, and Initial Host Infection
Phase 2 – Network intrusion, Remote Control, Lateral Movement, Data Discovery, Persistence
Phase 3 – Staging, Data Preparation and Data Exfiltration
Identify vulnerabilities, entry points, key individuals and key assets
Social engineering to gain access to privileged user account credentials
Attack Staging
Attack Launching
Initial Host Infection
Select targets individuals for Spear Phishing email
Weaponise content/ document with zero-day exploit
Compromise web site of interest to key individuals
Spear phishing email with embedded links to websites with zero-day malware downloads
Email with file attachments containing zero-day exploit sent
Zero-day exploit downloads and runs shell code
Local Host Compromise and Control
Lateral Movement
Persistence
Malware download and installation from C & C server
Credential Theft and Network Logon
Escalation of privileges
Download malware updates from C & C server
Malware injection into common processes
Trojan in reverse-connect mode. Connection to C & C server via TCP port 80.
Transmission of IP address, computer names in same domain, Windows password hashes
Steal username & password hash
Crack password hashes – dictionary attacks, rainbow tables
Compromised account has Network Logon user right on remote hosts
Connect to other computers 'net use',' net view', ' psexec.exe'
Additional tools for extracting password hashes and for cracking passwords downloaded
Computer containing higher privileged domain account compromised
Reuse stolen credentials to access server computers and services
Server with Domain administrator credentials is compromised
Establish access to staging servers at key aggregation points
Copy files from servers to staging servers
Data aggregated, compressed and encrypted for extraction
Password protected archive files exfiltrated via channel to outside staging server
Files deleted from staging server to remove traces of attack

Incident tree with typical events in APT-led data exfiltration

Phase 1 - Reconnaissance, Attack Staging, and Initial Host Infection

The first phase of an APT attack aims at compromising hosts with logon credentials of key individuals to start establishing a foothold in the network. This consists of four sub-phases.

Reconnaissance: The attackers research points of entry in the network, identify vulnerabilities, key individuals, their web-browsing habits and key assets in the organisation. As mentioned before, professional information on social media profiles of employees can provide clues of technologies in place and their potential vulnerabilities. From the same profiles, key individuals in the organisation such as top-ranking executives, IT administrators and individuals that can provide access to sought-after data within the organisation can be identified.

Attack Staging: After reconnaissance activities, the employees in the organisation who will be targeted, for instance, by spear-phishing emails, are selected. Office documents (Word, Excel, PDF) or other media (JPEG) that will be of interest to the target individuals are weaponised with zero-day exploits. In the RSA data breach, for instance, the attackers used a MS Excel Spreadsheet with embedded Flash content that contained a zero-day exploit for Adobe Flash vulnerability (CVE-2011-0609). This zero-day exploit, in this case, installed Poison Ivy, a well-known backdoor on the compromised host. Another aspect of attack staging is planting malware code in web sites of interest to the identified key individuals to facilitate drive-by-download attacks. Legitimate web sites visited by the targeted individuals can be compromised via cross-site scripting (XSS). The PHP.net web site compromise in October 2013 is an example of this attack vector. In this case, hackers managed to inject malicious JavaScript code into a file on the php.net site called userprefs.js. The code made requests to a third-party website that scanned visitors' browsers for vulnerable plug-ins and executed exploits that, if successful, installed a piece of malware. In the Google, Adobe data breach (Operation Aurora), for example, malicious web sites containing zero-day exploit code for Internet Explorer (that allowed remote code execution) were set up.

Attack Launching: Attack vectors aimed at gaining access to privileged hosts are launched. In various spear-phishing campaigns, email lures with embedded links to websites with malware downloads are sent to target individuals. Other individuals are targeted with emails with file attachments in common formats like Office or PDF. These attachments may include zero-day attack code targeting a previously unknown vulnerability or simpler attack code that takes advantage of vulnerabilities in unpatched/mis-configured software. Social engineering skills could also be used to gain access to privileged user account credentials.

Initial Host Infection: If the attack vectors are successful, then initial compromise of the target computers is achieved when shell code is downloaded and executed on them. For instance, shell code might be embedded in a JPEG image file targeting a vulnerability in MS Windows when handling .jpg image files (the Microsoft GDI+ Library JPEG Segment Length Integer Underflow vulnerability). This vulnerability allows a specially crafted .jpg file to cause Internet Explorer to execute arbitrary code stored in the .jpg file. The shell code will typically decrypt additional code and execute it. This code usually results in malware being downloaded from the C2 server and installed on the host.

Phase 2 – Network intrusion, Remote Control, Lateral Movement, Data Discovery, Persistence

In the second phase of an APT attack, the attacker seeks to establish a foothold in the network and then collect information on surrounding infrastructure, trust relationships and elements such as Windows domain structure. Subsequently, s/he will endeavour to expand control to other workstations, servers and infrastructure elements and perform data harvesting on them.

Remote Control: The downloaded malware provides remote access capabilities on the host machines to the attacker. The level of control afforded by the malware is sometimes near complete especially if kernel-mode rootkits have been deployed (these execute with the same privileges as the host's administrator). In the Operation Aurora (Google, Adobe) data breach incidents, for example, the remote code execution zero-day flaw in Internet Explorer resulted in the Trojan.Hydraq malware being installed on the host machines. The remote access capabilities provided by this Trojan include: controlling system processes and services, binary code download and execution, registry modification to ensure persistence, drive enumeration, file manipulation commands, computer restart capability, to name but a few. The malware enables the attacker to remotely control infected hosts with a C2 service that allows the attacker to remotely update malware, add new malware (encryption tools, etc.), and send commands to the host

Lateral Movement: At this stage, infected hosts download additional tools to steal data such as usernames and password hashes (e.g., gsecdump) or crack password hashes (e.g., Cain & Cabel). The aim is to gather valid user credentials (especially credentials with administrative privileges) and move laterally across the network, installing more back doors. The back doors allow the attacker to install bogus utilities and create a 'ghost infrastructure' for distributing malware that remains hidden in plain sight. With pass-the-hash attacks, the attacker tries to break into other hosts where the victim user has similar network logon rights. Access to computers requiring administrative access rights may be enforced using privilege escalation tactics. Such stepping-stone attacks allow attackers to jump from compromised access to a low interest account onto accounts with far more privileges before carrying out the end purpose of a multi-stage assault, normally the extraction of commercially or financially sensitive information. The attacker may also attempt to gain more control by discovering additional hosts within the target network and using network or other system-level vulnerabilities to infect those hosts.

Data Discovery: As the attackers traverse the network, they look for targets such as Active Directory (AD) and certificate PKI servers to gain account credentials and access privileges to confidential data within the network or the enterprise's cloud- based storage. They also try to find the location of intellectual property data repositories (e.g., source-code repositories in Operation Aurora).

Persistence: A key difference between traditional malware and an APT is the ability to persist. As opposed to traditional malware which is removed (by itself or by antivirus programs), APTs are designed to persist and go unnoticed for months. Rootkits or 'stealthkits' are popular tools for providing stealth to the operation of malware. Within each compromised host, they typically hide or 'cloak' files, directories and processes used by the malware and enable them to avoid detection by antivirus, anti-spyware and personal firewall applications. Another known technique is to inject the malware code in known processes (e.g. explorer.exe, iexplore.exe) or in system-level libraries. Some rootkits, of instance, infect system libraries such as the Windows Ntdll.dll to filter the results from system API calls to provide malware stealth. In addition, to survive detection by updated antivirus software, the APT may cause malware to request updates from the C2 servers in the form of new undetectable (code no known anti virus signatures).

Phase 3 – Staging-Server Selection, Data Preparation and Data Exfiltration

After the desired intellectual property has been located and access to it gained, the attackers prepare the data for extraction. To do so, they first select internal staging servers where the collected data will be aggregated. After data is moved to the staging servers, it is compressed and encrypted for extraction. The volume of data to extract and the degree of covertness desired determines the type of exfiltration method to use. In the RSA data breach, the attackers knew that security controls in place at RSA would eventually detect suspicious activity related to the exfiltration. They, therefore, chose to prioritise speed over stealth by using FTP to transfer many password protected RAR files from the RSA file server to an outside staging server. The files were subsequently pulled by the attacker and removed from the external compromised host to remove any trace of the attack. Other exfiltration channels such as DNS tunnelling or network stegnanography may offer less bandwidth but achieve a much higher degree of covertness.

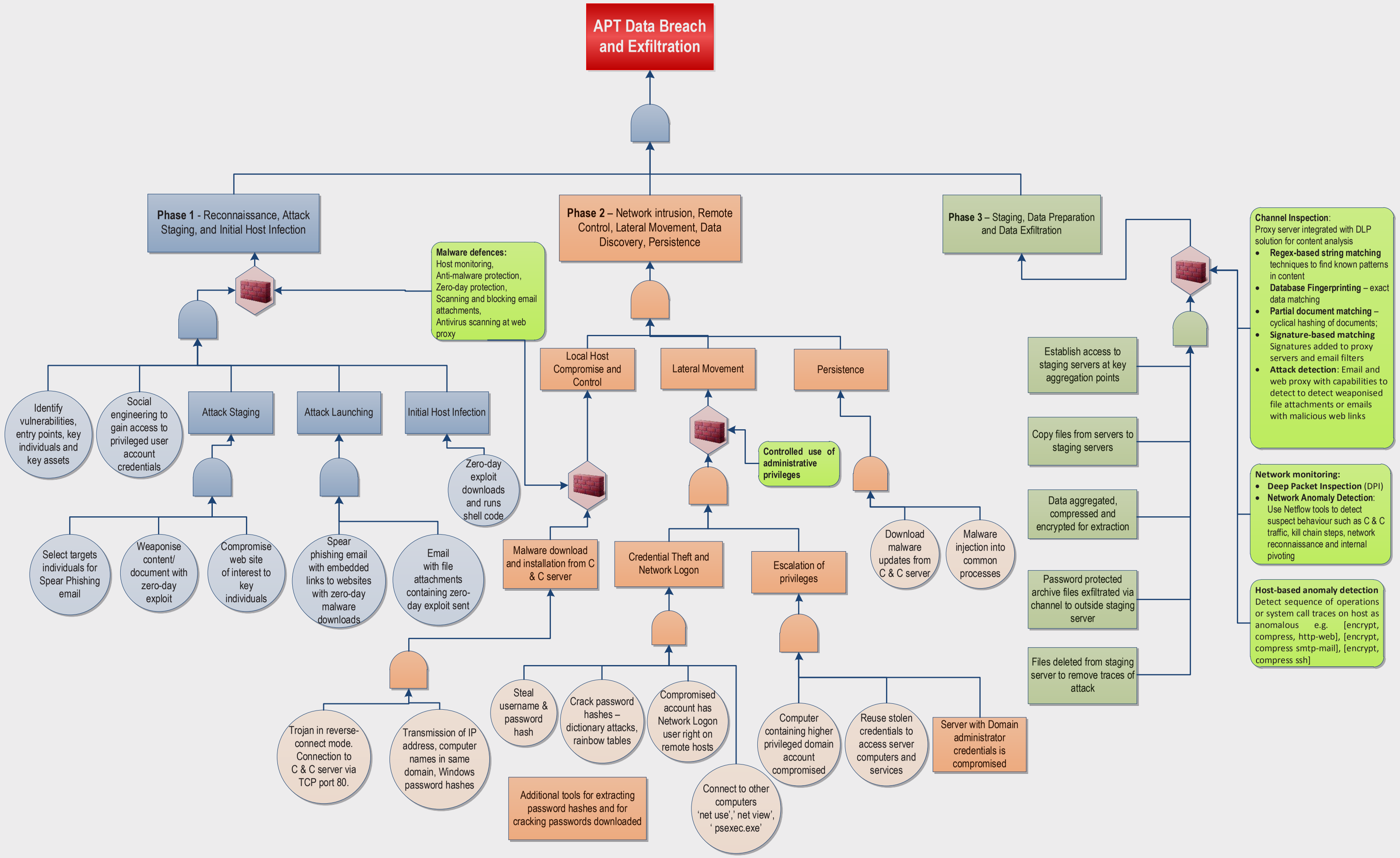
Although APT attacks are difficult to identify, the theft of data can never be completely invisible. Detecting anomalies in outbound data is perhaps the best way for an administrator to discover that the network has been the target of an APT attack.

Detecting and Preventing Data Exfiltration by APTs

A sound defence strategy against APT-led data exfiltration is to monitor inbound and outbound traffic for content, context, and data, preferably for both email and web communications. Networks with firewalls, IDS/IPS, and antivirus defences focus on inbound threat protection using signatures and individual defence analytics, and mostly ignore outbound communications. Traditional defences such as firewall and antivirus are necessary because they block known threat vectors; however, they are not sufficient and their limitations against APT techniques and targeted attacks must be recognised.

Known Channel Inspection

In addition to these traditional defences, we recommend performing content inspection on outgoing data in high-risk channels to detect data theft. Proxy servers for overt channels such as email, IM, web and FTP would enable the detection of sensitive data egressing the network; content analysers can look for matches in keywords, personally identifiable information, hashes, defined patterns or files flagged by signatures. The proxy servers can also provide an additional defence layer. A secure email proxy server could be upgraded with the ability to inspect for malicious web links and attachments to prevent initial infection. In addition to URL-filtering and virus scanning, a web proxy could perform real-time threat analysis to detect zero-day malware and non-binary-based malware (e.g., JavaScript) to prevent clients from being compromised. Malware signatures are often released by ethical hackers and security vendors upon the discovery of new variants. The web/mail proxies must have the ability to detect anomalous encrypted/SSL traffic and attachments to detect exfiltration of potentially sensitive data. If a reverse SSL proxy is made to operate a man-in-the-middle attack by, for example, installing a root certificate generated by the proxy into the browser CA list, then proxy analysis of the contents of a SSL/TLS transaction is possible.


APT Data Breach and Exfiltration
Phase 1 - Reconnaissance, Attack Staging, and Initial Host Infection
Phase 2 – Network intrusion, Remote Control, Lateral Movement, Data Discovery, Persistence
Phase 3 – Staging, Data Preparation and Data Exfiltration
Malware defences:
Host monitoring,
Anti-malware protection,
Zero-day protection,
Scanning and blocking email attachments,
Antivirus scanning at web proxy
Identify vulnerabilities, entry points, key individuals and key assets
Social engineering to gain access to privileged user account credentials
Attack Staging
Attack Launching
Initial Host Infection
Zero-day exploit downloads and runs shell code
Select targets individuals for Spear Phishing email
Weaponise content/ document with zero-day exploit
Compromise web site of interest to key individuals
Spear phishing email with embedded links to websites with zero-day malware downloads
Email with file attachments containing zero-day exploit sent
Local Host Compromise and Control
Lateral Movement
Persistence
Controlled use of administrative privileges
Malware download and installation from C & C server
Credential Theft and Network Logon
Escalation of privileges
Download malware updates from C & C server
Malware injection into common processes
Trojan in reverse-connect mode. Connection to C & C server via TCP port 80.
Transmission of IP address, computer names in same domain, Windows password hashes
Steal username & password hash
Crack password hashes – dictionary attacks, rainbow tables
Compromised account has Network Logon user right on remote hosts
Connect to other computers 'net use',' net view', ' psexec.exe'
Additional tools for extracting password hashes and for cracking passwords downloaded
Computer containing higher privileged domain account compromised
Reuse stolen credentials to access server computers and services
Server with Domain administrator credentials is compromised
Establish access to staging servers at key aggregation points
Copy files from servers to staging servers
Data aggregated, compressed and encrypted for extraction
Password protected archive files exfiltrated via channel to outside staging server
Files deleted from staging server to remove traces of attack
Channel Inspection:
Proxy server integrated with DLP solution for content analysis
• Regex-based string matching techniques to find known patterns in content
• Database Fingerprinting – exact data matching
• Partial document matching – cyclical hashing of documents;
• Signature-based matching Signatures added to proxy servers and email filters
• Attack detection: Email and web proxy with capabilities to detect to detect weaponised file attachments or emails with malicious web links
Network monitoring:
• Deep Packet Inspection (DPI)
• Network Anomaly Detection: Use Netflow tools to detect suspect behaviour such as C & C traffic, kill chain steps, network reconnaissance and internal pivoting
Host-based anomaly detection
Detect sequence of operations or system call traces on host as anomalous e.g. [encrypt, compress, http-web], [encrypt, compress smtp-mail], [encrypt, compress ssh]

Barriers that can be deployed to counter data exfiltration by APTs

Network Monitoring

Some examples of malicious outbound behaviour in APTs are C2 traffic, requests to dynamic DNS hosts, requests to known 'bad' web locations, movement of sensitive files that should never be sent outside the organisation (e.g., MS Windows' Security Accounts Manager database), and the use of proprietary encryption. In order to combat APTs, it is imperative that organisations know what is going on within their internal networks to fill in the gaps left by perimeter security solutions. The first step towards network monitoring is to turn on the logging features in the various software components in the network such as the Web Proxy server, the DNS server. This would allow the detection of particular behavioural patterns. Thus, HTTP requests sent to suspicious domain names can offer an indication of exfiltration towards an unknown destination. Periodic communication with an unknown destination server followed by large data transfers or payload downloads can provide a strong indication of malware activity. In addition to network-level logging, other measures include network monitoring solutions such as Deep Packet Inspection to detect unauthorised movement of sensitive files outside the organisation's network (via signature-matching, for example).

We recommend that organisations use network-based tools to monitor and control data flows within the network. Any anomalies that exceed the normal traffic patterns should be noted and appropriate action taken to address them. For instance, Netflow or sFlow tools provide system administrators with visibility of communication flows in their network. By leveraging flow data and the sophisticated behavioural analysis that is provided by these tools, administrators can detect the various steps that sophisticated attackers take to infiltrate a network, including network reconnaissance, covert C2 communications and internal pivoting. As mentioned previously, flow monitoring capabilities by Netflow solutions would enable anomalies to be detected in observed traffic patterns.

It is also advisable that all DNS queries be logged and DNS queries on unusual domain names (for example, outside the Access Control List) or known malicious C2 domains blocked.

Host-based Data Encryption and Anomaly Detection

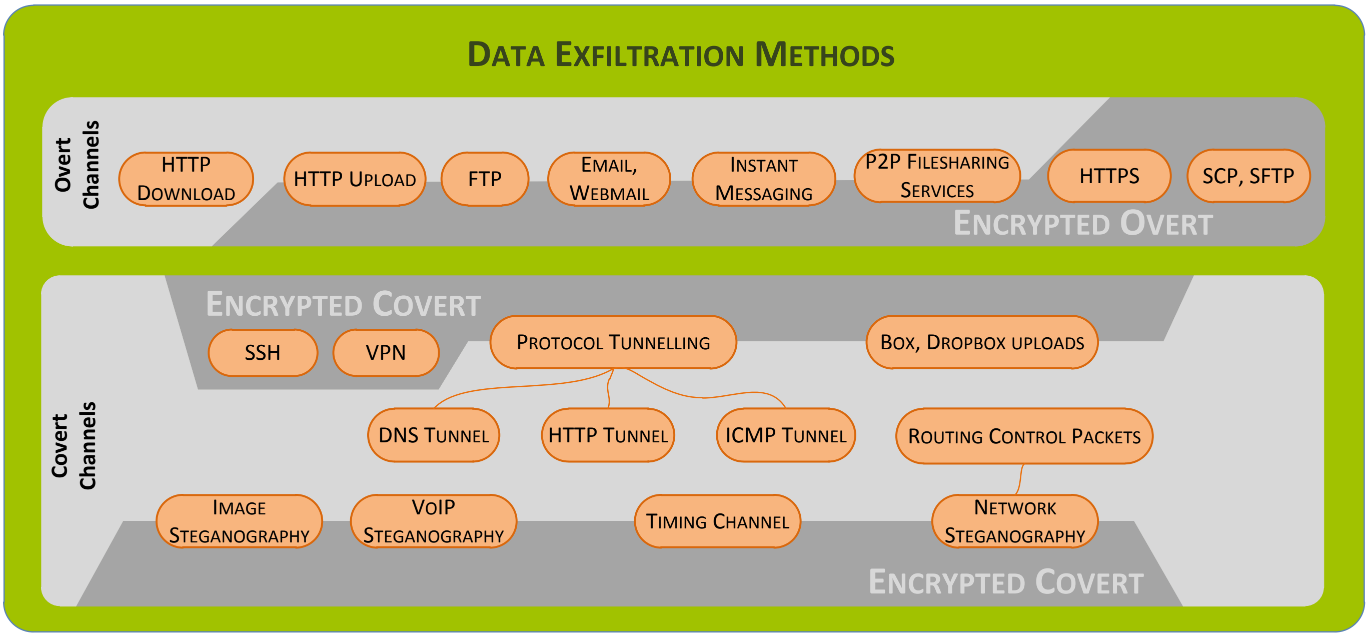
Disk and database encryption to prevent unauthorised access to sensitive data can help mitigate threats posed by APT-based data exfiltration attacks. Host-based monitoring to detect anomalous behaviour due to malware activity, importation of tool sets or internal network reconnaissance operations by attackers is another measure to detect and disrupt APT attacks. Advanced host monitoring agents hook into the operating system to, for example, detect traces of attacker activity from the system call table (e.g., scanning, brute-forcing, and service exploitation).

Malware Defences

As malware is often a weapon of choice in many data exfiltration campaigns, actions to neutralise malware on the compromised host can be effective when applied in tandem with the other techniques described above. In particular, the following malware defence mechanisms are recommended:

- Employ automated tools to continuously monitor workstations and servers for active, up-to-date anti-malware protection with antivirus, anti-spyware, personal firewalls, and host-based IPS functionality. All malware detection events should be sent to enterprise anti-malware administration tools and event log servers.
- The endpoint protection tools should also include zero-day protection.
- All email attachments entering the organisation's email gateway must be scanned and blocked if they contain potentially malicious code or file types unauthorised within the organisation. Email and web content filtering should be applied at the network perimeter through the use of proxy servers.
- Toolkits such as Data Execution Prevention (DEP) and Enhanced Mitigation Experience Toolkit (EMET) can be deployed to provide sandboxing (e.g., run browsers in a Virtual Machine), and other techniques that prevent malware exploitation.
- Access to external email systems, instant messaging services, and other social media tools should be blocked.
- Network-based anti-malware tools should be utilised to analyse all inbound traffic and filter out malicious content before it arrives at the endpoint.
- Continuous monitoring should be performed on all inbound and outbound traffic. Any large transfers of data or unauthorised traffic should be flagged and, if validated as malicious, the computer should be moved to an isolated VLAN.
- DNS query logging should be enabled to detect hostname lookup for known malicious C2 domains.

Data Exfiltration Methods



Technical methods for extracting data from an organisation's systems range from those commonly seen in attacks to niche vectors and cutting-edge methods for covertly moving information past current detection systems. These methods largely fall into two categories: those using overt channels and those making use of covert channels and other C2 mechanisms.

Overt Channels

These channels are the ones that are generally used by any computer user wishing to transfer files between locations. This makes such vectors obvious targets for scrutiny, but also means that corporate networks generally provide for some freedom in their use[9]. These channels commonly feature in literature on threat taxonomies. Here we discuss them with particular reference to remote attackers and data exfiltration.

HTTP Download

Perhaps the most direct method of data exfiltration for a remote attacker is manipulating a public-facing server into disclosing non-public information in response to a HTTP POST or GET request. The most widespread class of malicious attacks, SQL injection attacks, may use this method to expose the contents of databases connected to public websites, simply downloading the result of a malicious query in the same manner they would download a webpage. This is often also a precursor to the attacker using the extracted information to gain greater access to the target system.

SQL injection is a type of Web application security vulnerability in which an attacker is able to submit, as part of input data, a database SQL command that is executed by a Web application, exposing the back-end database. By doing so, the attacker attempts to intentionally access/steal information assets without authorisation by circumventing or thwarting logical security mechanisms. The vulnerability of public-facing servers to attacks like SQL injection is well-known, and the steps involved in identifying and exploiting such vulnerabilities are well-documented [18;52;70] and in many cases trivially automated for attackers.

Upon detecting SQL injection (SQLi) vulnerabilities, attackers commonly extract and/or bypass authentication or escalate privileges to extract other sensitive information. An attacker can identify injectable parameters, perform database fingerprinting, determine the database schema, extract data values from the database, and add or modify data, to name a few SQLi exploit capabilities. The results can then be shipped over the network using HTTP requests. If the attacker achieves console access or access to scripting facilities such as PowerShell, s/he has all of the facilities available in the host operating system to gather, package and deliver the data just as a user would on the system. This affords an attacker the same luxuries for evasion and stealth with regards to data exfiltration that would ordinarily be reserved for an attacker that had completely compromised the host system.

HTTP or FTP Upload

Attackers with access to their target system may well make use of file transfer systems familiar — and usually available — to any office worker. HTTP traffic often carries uploaded files, and attackers can make use of a variety of freely available file-hosting sites to store data for later retrieval. File Transfer Protocol connections can be used in a similar manner to reach drop-sites, or else used to directly transfer files between the target and a machine controlled by the attacker.

A rising class of exfiltration threats are those that target not the usually-monitored servers of an organisation, but rather the personal computers of persons likely to have access to targeted information or credentials. The methods available are numerous, ranging from manipulating the user into volunteering the information in a phishing attack[22] to manipulating their software through any number of technical exploits such as cross-site scripting[28; 30] or exploits which directly compromise their machine and then report back, sometimes by embedding information in innocuous web traffic[31].

Email

The various email protocols are near ubiquitous in modern business networks, and easily capable of carrying small-to-mid size files. Attackers can mail files directly to inboxes controlled by themselves — the modern webmail environment making creating temporary accounts a simple matter — or else make use of public mailboxes as drop-sites from which files can later be retrieved[9;22].

Instant Messaging Applications

Many businesses make use of Instant Messaging (IM) applications for communications purposes. Even when these applications are not officially authorised by IT departments, users may make unofficial use of them on corporate networks. Due to the potential number of different protocols over which IM could be carried, and the often in-built protocol tunnelling systems offered by clients, blocking IM traffic can be a difficult proposition. Attackers using an IM application to exfiltrate data may use the text body of outbound messages, but are more likely to make use of in-built file transfer systems or else a client's streaming capability, which most modern IM protocols offer [56;57;58;59;60].

Covert Channels

Encrypted Channels

As many of the overt channels detailed above may be monitored for exfiltration attempts, attackers evade the detection of network-based monitoring by concealing the data they are removing. The most direct means of doing this is to encrypt the data before sending it over any overt exfiltration channel. Due to the widespread adoption of SSL/TLS, the ability to initiate secure (that is, encrypted) connections is common and usually necessary in most modern networks. Without specific countermeasures in place, network monitors would not be able to detect the contents of encrypted data packets being delivered via HTTPS or common utilities like SCP or SFTP[22], and thus would not be able to identify sensitive data being extracted.

Instead of or as well as TLS encryption, attackers may use asymmetric key encryption to disguise data being sent out of the network. In such encryption schemes, one key is used to encrypt the data and only another key mathematically linked to it may unlock the cipher. This scheme is useful for attackers uploading data to public drop sites, as they can leave the encrypted data open to the public, but only they have the key to access it. This scheme also has the advantage that it is provably impossible for analysis to reveal whether a sophisticated attacker making use of it has in fact exfiltrated secret data[88], leading to uncertainty in the aftermath of an event.

It is worth noting that, as most data leakage protection (DLP) systems, rely on simple pattern matching or keyword-lists, even the most rudimentary encryption can be sufficient. In an office environment where the malicious attacker has no ability to install sophisticated encryption software, simple yet effective ciphers can be implemented with common utilities like Microsoft Excel[13].

Steganography

Steganography is the art of hidden writing. Where cryptography seeks to hide the content or meaning of a message, steganography hides the message itself, perhaps by embedding it in another message. Typically, modern steganography works by identifying either redundant space within innocuous files or unused fields in common communications protocols (including the ubiquitous TCP/IP), and then encoding the message into these overlooked areas[54]. Cryptography can be used alongside steganography to encrypt the hidden data, posing a dual challenge to any exfiltration detection system – firstly to detect that a message is being hidden, and secondly to discover what that message is.

Steganography is a large field, and a complete survey of methods would be outside the scope of this report. However, a review of literature reveals that steganographic technology is easily available to attackers or office workers in a variety of commercial or free technology suites[54], that alongside image formats commonly used for steganography and common office document formats are highly suitable for embedding hidden information[16]. A good example of the variety of stenographic solutions available to attackers is VOIP (Voice-over-IP) steganography that, in the past few years, has made a transition from proof-of-concept ideas in the lab to tools that can be exploited for data exfiltration.

VoIP steganography conceals secret messages/data within VoIP streams without severely degrading the quality of calls. As VoIP is very popular, its usage will not raise suspicions, i.e., it will not be considered as an anomaly in itself. In fact, the more frequent the presence and utilisation of such carriers in networks, the better their masking capacity, as hidden communications can pass unnoticed amongst the bulk of exchanged data. Potentially high steganographic bandwidth can be achieved using this class of methods. For example, during the conversation phase of a G.711-based call, each RTP packet carries 20 ms of voice; in this case the RTP stream rate is 50 packets per second[107]. Thus, even by simply hiding 1 bit in every RTP packet one can gain a relatively high steganographic bandwidth of 50 bit/s. VoIP sessions involve the combined use of a variety of protocols. Thus, many opportunities for hiding information arise from the different layers of the TCP/IP stack (e.g. IP, TCP, UDP, RTP header fields). Apart from using network steganography, clandestine data exfiltration can also be enabled by employing steganographic methods applied to the users' voice that is carried inside the RTP packets' payload, by utilising audio watermarking techniques.

As VoIP is a real-time service, it induces additional strict requirements for steganographic methods but also simultaneously creates new opportunities for steganography (e.g., utilisation of excessively delayed packets that are discarded by the receiver without processing because they cannot be considered for voice reconstruction).

Five popular techniques have emerged from this class of exfiltration methods:

Steganophony. This type of exfiltration method entails hiding data inside each voice payload packet but not to such an amount that it degrades the quality of the sound. A number of audio watermarking techniques (Least Significant Bit [107], Quantisation Index Modulation[107], etc.) can be utilised in real-time communication over a VoIP service to embed covert data (thus creating a covert channel) inside the audio content. For these types of covert channels, the bandwidth available depends mainly on the sampling rate and the type of audio material being encoded. The covert data rate has to be limited as to avoid causing voice quality deterioration and increased risk of detection. With a sampling rate of 1 KHz, an audio watermarking algorithm such as LSB has been empirically found to provide a covert bandwidth of 4kbps[107]. This is ample enough for common data exfiltration purposes.

Network steganography. A VoIP solution uses signalling protocols for such as SIP and H.323 for brokering connections for voice calls. After a connection has been established, transport protocols such as RTP (and the underlying UDP) are used to provide end-to-end network transport functions suitable for applications transmitting real-time audio. Other supplementary protocols such as RTCP are used to monitor the quality of service during the call and convey information about participants in the session. In network steganography, covert data is inserted into redundant fields for the above-mentioned protocols and extracted on the receiving side.

Intentionally delayed Audio Packets Steganography. This technique exploits the fact that for real-time multimedia communication protocols like RTP, excessively delayed packets are usually dropped by the receiver and are not used for current voice reconstruction at arrival time. Thus some selected audio packets are intentionally delayed before transmitting; these packets are the ones that contain the secret payload. Normal VoIP receivers, which are unaware of the covert channel enabled by the steganographic procedure, discard these packets. For receivers aware of the covert channel, the hidden data can be retrieved from the packet payload with little chance of detection.

Masquerading stolen data as VoIP traffic. This method initially entails sniffing the network to observe recurring patterns of calls, and user identifications (to be later used when initiating the SIP call). After an initial pattern can be mapped out, the exfiltration method involves encoding the data to be exfiltrated from its binary format to audio. The encoding scheme maps byte values from the data stream to a corresponding scale of audio tones using 16 distinct octaves on the human audible frequency range (20Hz to 20,000Hz), thereby making the covert data transmission appear as a normal VoIP call. Once encoded, the final output can then be played back on an opened SIP call that can be made to almost any number outside the organisation (for example a Google voice account's voicemail) for later decoding back to the original binary data.

HICCUPS[89]. This method calls for inserting extra and deliberately malformed packets within the VoIP flow. They will be dropped by the receiving application, but can be picked up by other devices on the network that have access to the entire VoIP stream. When the receiver's computer gets a packet, it checks for errors using that packet's checksum. Normally, if the checksum is wrong, the computer discards that packet. But if a terminal has the correct steganography program installed, it doesn't discard these intentionally wrong checksums—instead, it will identify that these are precisely the data packets to scan for steganograms. This exfiltration method is often exploited in a wireless context and provides a fast covert channel for exfiltration. In an IEEE 802.11g network with a transmission capacity of 54 megabits per second, with 10 terminals and a 5 percent rate of corrupted frames, the bandwidth attainable for covert data transmission has been found to be higher than 200 kilobits per second. This method requires a wireless card that can control frame checksums. Detecting HICCUPS is not straightforward. One would need some way of observing the number of frames with incorrect checksums. If the number of those frames is statistically anomalous, then one might suspect the transmission of hidden information. Another way of detecting HICCUPS would be to analyse the content of those dropped—and therefore retransmitted—frames in order to detect the differences between the dropped and retransmitted frames. Major differences in these frames would provide an obvious clue to malicious activity.

Protocol Tunnelling

An exfiltration threat related to steganography is protocol tunnelling – the practice of embedding certain network traffic for an untrusted application within permitted traffic. Such tunnelling is highly effective at evading blocked ports and allowing untrusted applications (like P2P filesharing software) to evade boundary defences[66]. An example of this is DNS tunnelling .

The Domain Name System (DNS) is a protocol used by web-based and email applications to request translation of domain names to IP addresses; the IP addresses are then used by the protocol stack on the host computer to route data packets to the intended destination computer. Along with HTTP, DNS resolution protocols are the most commonly used communication protocols in an enterprise and are therefore quasi-guaranteed to be allowed through an organisation's perimeter firewall. However, as DNS is not intended for data transfer, it has generally been overlooked as a threat for malicious communications or for data exfiltration. Many organisations are wide-open to attacks via this exfiltration vector since they tend to focus resources on monitoring web or mail traffic where traditional exfiltration vectors tend to operate.

DNS tunnelling is the misuse of the DNS service to tunnel another protocol through. A DNS tunnel can thus be used for C2 or data exfiltration. At the 2012 RSA conference[90], DNS-based Command and Control of malware was described as one of the six most dangerous new attack vectors. This technique was shown to have been used in recent data breach incidents involving the theft of millions of accounts[90]. In terms of achievable data rates, it has been shown that DNS tunnelling can achieve bandwidth of 110 KB/s (Kilobytes per second) with latency of 150 ms [91].

There is a wide variety of tools (DeNiSe, dns2tcp, DNScat, iodine, to name but a few) available for tunnelling over DNS; they have been mostly created with the intent of obtaining free Wi-Fi access from paid Wi-Fi sites with a captive portal for HTTP, but free-flowing DNS. Each tool may use different DNS request/response types; for example, record types such as 'A', AAAA, KEY, TXT, CNAME may be used to hold the covert data.

Although, their DNS request/response types may differ, the mechanisms employed by these utilities for DNS tunnelling are based on four core techniques: **1)** a controlled domain or subdomain, **2)** a server-side component, **3)** a client-side component, and **4)** data encoded in DNS payloads.

The controlled domain is used to define the authoritative name server for that domain or subdomain. Any DNS requests containing that domain or subdomain will be routed through the hierarchical DNS system till it reaches the DNS server that serves domain name resolution requests for that particular domain/sub-domain. In the DNS tunnelling exfiltration vector, the server-side component known as the DNS tunnel server, is the authoritative name server for the controlled domain and will typically run on an internet-accessible server controlled by the tunnel user. The client-side component hosts the other end of the tunnel. This could be an endpoint in a security-controlled enterprise environment. The tunnel could be used to communicate past the security controls and allow communication between the controlled endpoint and an arbitrary host on the Internet.

To perform exfiltration, the client-side component initiates a DNS request for which the DNS tunnelling server is the authoritative name server with an encoded message. This DNS query is sent to the organisation's internal DNS server. The internal DNS server forwards the request to the next DNS server; the DNS query message is forwarded through the hierarchy of DNS servers until it reaches the authoritative DNS server under the attacker's control. The attacker can then decode the exfiltrated data (or request) and send back an encoded response. The client component will receive the response and extract the encoded data/command in the message. A client could send an 'A' record request where the data is encoded in the hostname: `MRZGS3TLEBWW64TFEBXXM YLMORUW4ZI.x.example.com.` The server could then respond with an answer as a CNAME response: `NVWW2IDPOZQWY5DJNZSQ.x.example.com.`

Although, the efficiency of DNS Tunnelling as an exfiltration vector is limited by DNS protocol restrictions and the small number of encoding schemes available, it is still the preferred method by attackers in heavily filtered environments where outbound connectivity is restricted. In particular, some malware programs have used DNS tunnelling. For example, Feederbot[92] and Moto[93] are both examples of malware that use DNS TXT records for command and control. A number of exploit kits typically offer multiple exfiltration channels including DNS tunnelling to be used in the most-filtered environment. For instance, Squeeza is an SQL injection tool that supports three exfiltration channels: http errors, timing channels and DNS tunnelling.

Timing Channels

This method of exfiltration usually appears in literature describing threats but rarely in literature aiming to detect or prevent exfiltration, yet it can be exploited by sophisticated attackers for data exfiltration. A timing channel is an extremely subtle form of hidden channel which works by sending an innocuous packet to an external recipient at a particular time, such that the time delay between packets represents a particular byte value[22]. Such a vector is very difficult to detect, as any traffic could potentially be carrying a timing channel, and the information being communicated is not embedded in the packets themselves, merely in the delay between them.

Wireless Networks

An increasingly relevant and sometimes overlooked exfiltration vector is the sniffing of wirelessly broadcast traffic. With many wireless networks still insufficiently secured[65], and businesses making more and more use of wirelessly-connected laptops, tablets, smartphones and other devices, the threat of attackers passively listening to an organisations' traffic is a very real one, and many connected devices leak information[23]. Such broadcast-interceptions are not limited to typical wireless networks. In 2009, it was discovered that military adversaries of the United States in Iraq were able to access the video feeds of Predator drones by simply listening on the correct channel[74]. Any medium which broadcasts information to an unknown number of users – be it a satellite transmission or a corporate wireless network – should be handled carefully when it comes to confidential information.

Virtual Machine Vulnerabilities

Modern businesses are making greater use of virtual machines hosted by a third party. However, virtualised computing infrastructure carries its own risks with regards to data exfiltration – including the risk of a service provider being able to access your data, even if it is encrypted[27] – and a number of articles in literature point out covert channels available to attackers targeting such systems.

These threats mostly come from co-residency, where a malicious virtual machine is set up on the same physical machine as a target virtual machine. Such a situation can be detected by the malicious machine[10;6] using watermarks to identify the target, and then a number of covert channels can be employed to transfer data from one virtual machine to the other, including exploitation of the physical machine's cache[95] and memory bus[94], as well as utilising timing channels over network sockets[69].

Case Study: Data Exfiltration by RAM Scraping

Cybercriminals have been infecting consumer PCs with information-stealing trojans for years in the hope of harvesting potentially lucrative information such as credit card data or account credentials. Recently however, there has been a growing trend to target Point of Sale (POS) systems. Malware that infects POS terminals can be one of the most efficient ways to carry out payment card fraud because it targets machines with access to large amounts of the required data. Instead of going through the trouble of infecting tens of thousands of consumer PCs or physically installing a skimmer, an attacker can achieve the same results by targeting just a few POS systems with specially crafted malware.

A recent example is the Dexter custom-made malware that harvests credit card data by obtaining memory dumps (the memory contents of programs) from POS software and parsing these to look for payment card data. Uncovered in 2012 by Seculert, Dexter has been involved in a string of data exfiltration attacks in POS systems. Around the period preceding December 2012, Dexter was used to infect hundreds of point-of-sale computers at big-name retailers, hotels, restaurants, and other businesses in about 40 different countries. A second string of cyber-attacks was reported by the Payment Association of South Africa (Pasa) in October 2013 where payment card systems of thousands of shops, restaurants and hotels had been compromised.

The past ten years are littered with other examples of POS compromises affecting government agencies and businesses in almost every sector including hotels, restaurants, retailers, etc. Many of the most egregious data breaches are never publicly disclosed, the only externality is an unexpected new payment card in the mail. The threat is real and it will continue unabated until the technological barriers to entry are raised significantly. Payment card track data is very valuable to cyber criminals as it is easy to monetise. Consumers' payment card track data with or without PIN is one of the most sought after criminal commodities in the online marketplace.

Anatomy of RAM Scraping Data Exfiltration

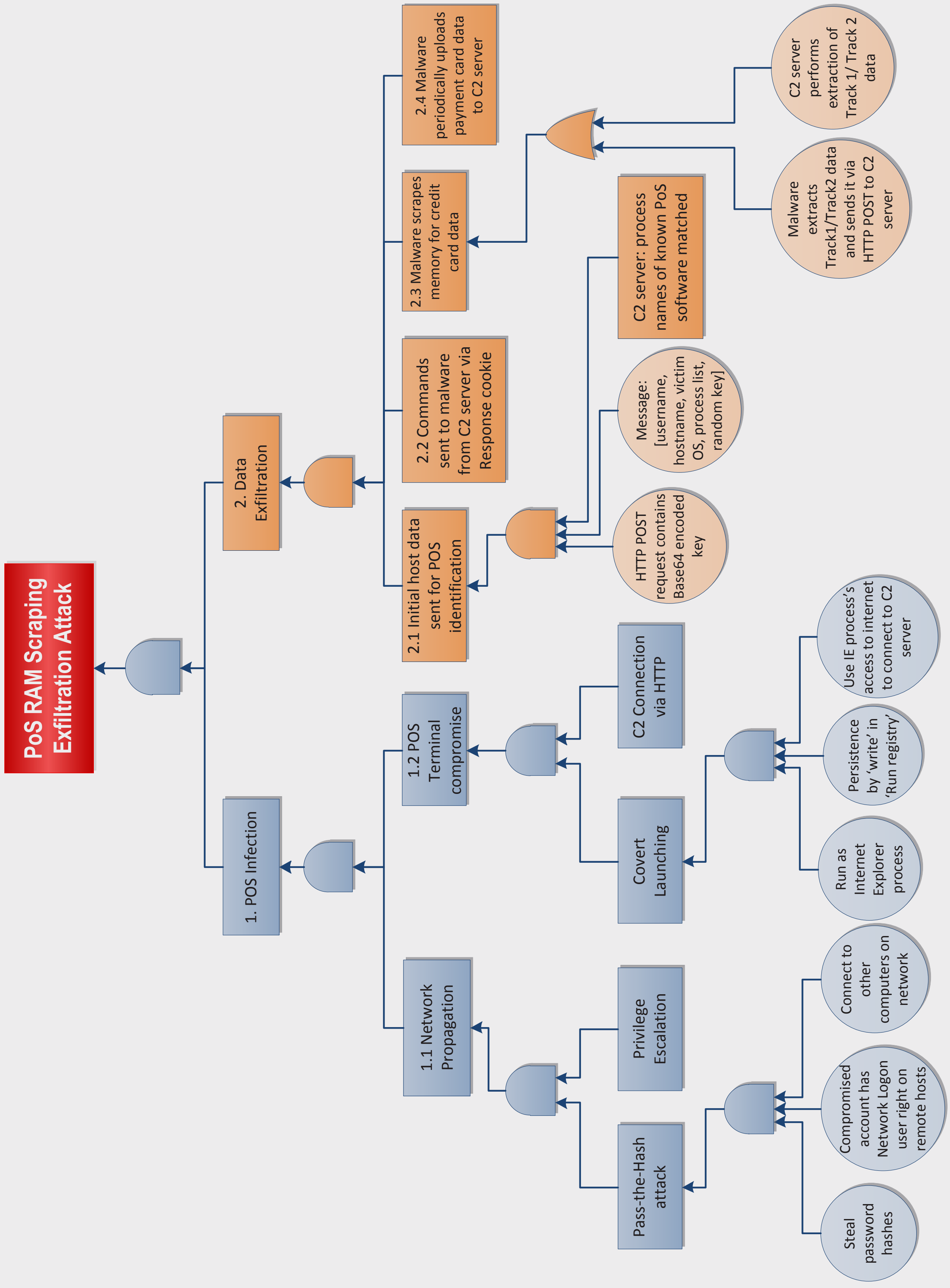
A variety of infection vectors are possible including malicious email attachments or drive-by-downloads if email/internet access is enabled on POS systems. Drive-by-downloads involve compromised websites taking advantage of security flaws of the POS system such as out-of-date browsers and operating systems with zero-day vulnerabilities (for example, Java, PDF or JPEG buffer-overflow exploits) to install rootkits, which are subsequently used to download compressed files containing the malware. Targeted attack on the infrastructure of particular organisations where the attackers either compromise company servers or infect POS terminals through physical means (for example, plugging a USB key into an unattended POS terminal) can also provide access. If the target is a large retail company, the attacker(s) can assume that it is a Level 1 PCI compliant merchant. Therefore in downloading the malware suite of tools, probably in the form of a single compressed file, the attacker will choose a communication protocol (FTP, HTTPS, RDP, VNC, DNS, etc.) that has a higher chance of connection success without encountering egress proxy issues.

Alternatively, as shown by the incident tree modelling the attack, the attacker can propagate the malware through the network using pass-the-hash attacks and privilege escalation to gain access to POS terminals. Using OS fingerprinting techniques, the attacker targets only systems that run Windows or other compatible platforms. Once copied on a victim computer, a typical scenario involves the malware injecting code in a known process (for example, as a DLL into the Internet Explorer process iexplore.exe, on Windows) as a way to conceal itself and to bypass firewall restrictions since Internet Explorer is allowed network access by most firewalls. On Windows-based POS platforms, malware such as Dexter ensures persistence to system restarts by writing to the 'Run' registry key. As soon as it is installed and launched on a host computer, the malware connects to its C2 server via HTTP by sending a message via HTTP POST containing host details such as username, platform configuration (operating system and processor type) and the list of running programs (process names). It may use an overt exfiltration channel with encryption to avoid detection by traffic inspection.

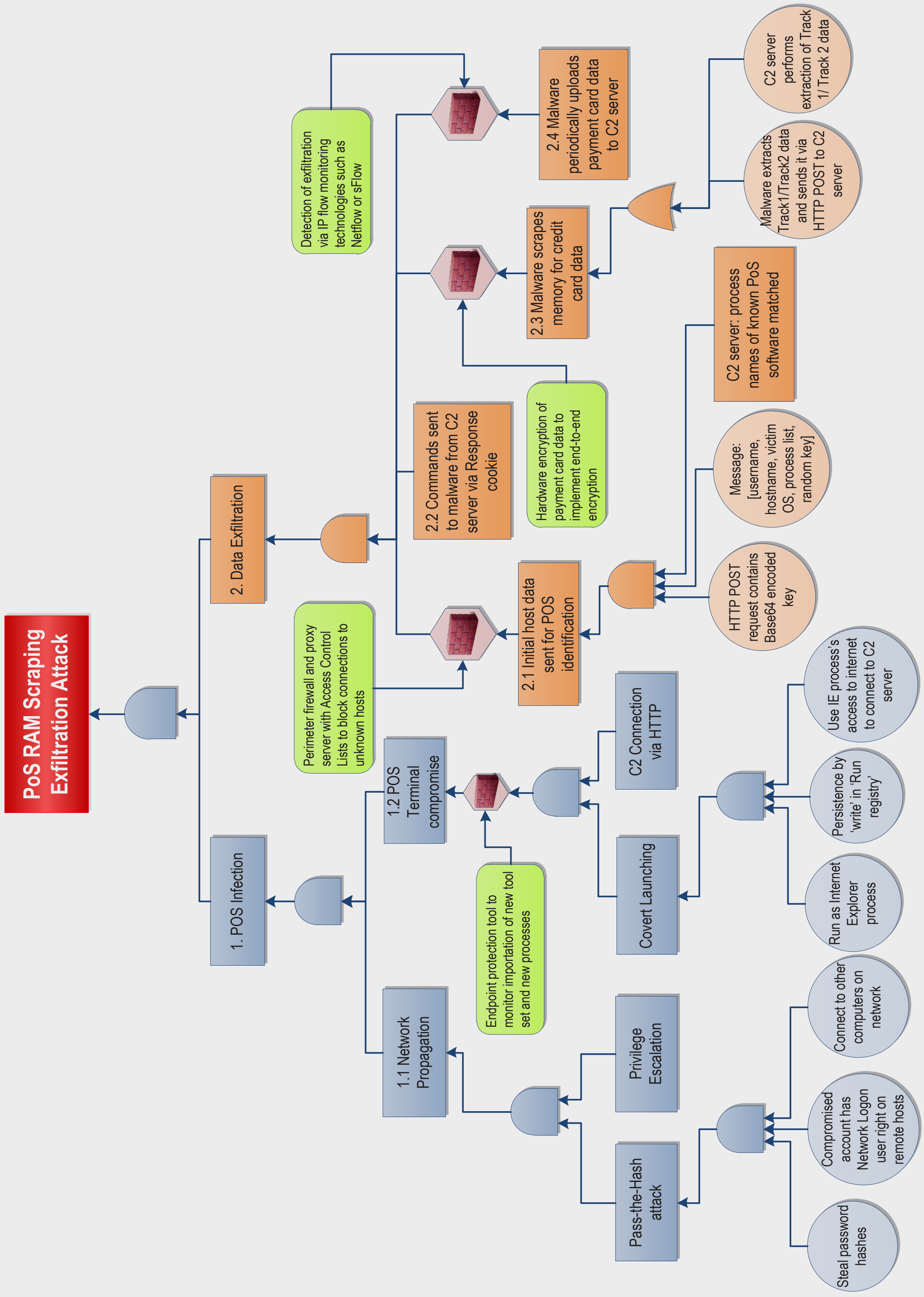
Upon receiving the list of process names from a malware instance, the C2 server determines the type of POS software that is running on the victim computer. After a successful match, instructions are sent to the malware to perform memory dumps of the identified POS software process and parse the memory contents to extract payment card data. Dexter for instance, will parse the memory contents to extract Track 1/Track2 data of payment cards. Other commands emanating from the C2 server could configure the time instances when the malware should scrape memory for such data.

Detecting and Preventing RAM Scraping Data Exfiltration

In any enterprise network, focusing exclusively on intrusion prevention is a lost cause as the opportunities for attackers to gain access to the network abound. S/he may buy a compromised host in the network from a bot herder, craft a reasonable phishing email, or use a remote vulnerability in the content management system (CMS) of the marketing department's most visited web destination for iframe-redirection to his/her Metasploit server. By looking at the chain of events leading to the exfiltration in the incident tree, it is possible to propose barriers (as illustrated by the inhibit barriers in the figure) in the form of detection/prevention mechanisms that will alert or block the occurrence of some of the events and thus disrupt or, at the very least, detect the propagation and operation of the malware memory scraper.


PoS RAM Scraping Exfiltration Attack
1. POS Infection
2. Data Exfiltration
1.1 Network Propagation
1.2 POS Terminal compromise
2.1 Initial host data sent for POS identification
2.2 Commands sent to malware from C2 server via Response cookie
2.3 Malware scrapes memory for credit card data
2.4 Malware periodically uploads payment card data to C2 server
Pass-the-Hash attack
Privilege Escalation
Covert Launching
C2 Connection via HTTP
Steal password hashes
Compromised account has Network Logon user right on remote hosts
Connect to other computers on network
Run as Internet Explorer process
Persistence by ‘write’ in ‘Run registry’
Use IE process’s access to internet to connect to C2 server
HTTP POST request contains Base64 encoded key
Message: [username, hostname, victim OS, process list, random key]
C2 server: process names of known PoS software matched
Malware extracts Track1/Track2 data and sends it via HTTP POST to C2 server
C2 server performs extraction of Track 1/ Track 2 data

Anatomy of Data Exfiltration by RAM Scraping


PoS RAM Scraping Exfiltration Attack
1. POS Infection
2. Data Exfiltration
1.1 Network Propagation
1.2 POS Terminal compromise
Pass-the-Hash attack
Privilege Escalation
Covert Launching
C2 Connection via HTTP
Steal password hashes
Compromised account has Network Logon user right on remote hosts
Connect to other computers on network
Run as Internet Explorer process
Persistence by 'write' in 'Run registry'
Use IE process's access to internet to connect to C2 server
Endpoint protection tool to monitor importation of new tool set and new processes
Perimeter firewall and proxy server with Access Control Lists to block connections to unknown hosts
2.1 Initial host data sent for POS identification
HTTP POST request contains Base64 encoded key
Message: [username, hostname, victim OS, process list, random key]
C2 server: process names of known PoS software matched
2.2 Commands sent to malware from C2 server via Response cookie
Hardware encryption of payment card data to implement end-to-end encryption
2.3 Malware scrapes memory for credit card data
Malware extracts Track1/Track2 data and sends it via HTTP POST to C2 server
C2 server performs extraction of Track 1/ Track 2 data
2.4 Malware periodically uploads payment card data to C2 server
Detection of exfiltration via IP flow monitoring technologies such as Netflow or sFlow

Barriers to detect, prevent and mitigate RAM scraping exfiltration

So assuming that the attacker is already in the network, his/her next step would be to locate payment card data and install the memory scraper malware on the POS terminals. Hence, a first set of countermeasures could aim at detecting/ mitigating network enumeration, credential brute forcing, privilege escalation and pass-the-hash attacks. Another countermeasure is to restrict Internet connections on POS terminals by deploying a perimeter firewall and a proxy server such as Squid that uses Access Control Lists to only allow connections to permitted hosts.

If the attacker is already equipped with the payment processing network segment location and the requisite credentials to access it, then the most useful indicators to focus on for detection are **1)** the importation of a tool set on a host, **2)** a new process running on the POS terminal, and finally **3)** the exfiltration of messages or compressed files with uniform size and frequency. Tools can be deployed to trigger alerts on this chain of events or on any specific event. Technologies such as Netflow and sFlow, for instance, monitor IP flows in the network and can use historical net flow data to detect anomalous data transmissions. Many Netflow tools provide a number of pre-installed, well thought-out alerts together with the possibility of creating new alerts. To look for payment card data exfiltration by POS memory scrapers, one could create an alert rule based on repeating byte count patterns above a reasonable threshold and potentially add additional criteria for periodicity.

Such tools also include a rule for 'Suspect Data Loss' which trigger alerts on large disparities between the quantity of inbound and outbound packets. Although, a lot of legitimate traffic falls into this category, not just exfiltration, historical flow data on the network is analysed to create a behaviour baseline. This behaviour model can then be used by flow monitoring tools to alert on flow sessions that qualify as suspect outliers. Furthermore, application and process change detection should be in effect on all payment card processing systems. Any change on the end point or multiple end points should be cause for immediate analysis.

Finally, the opportunity of using memory scrapers to steal payment card data arises because of the fleeting moment the data from the payment card's magnetic stripe is read into memory at the POS terminal before being encrypted and sent to the card payment processor. End-to-end encryption of the payment card data is necessary right from the moment the card details are read at the POS terminal to when the payment is authorised by the payment processor.

Detecting Data Exfiltration

Organisations can counter the threat of data exfiltration from their systems by utilising a range of methods including those for detecting ongoing attempts as well as ones for identifying which people or system components were involved in a breach after the fact. Timely detection of exfiltration attempts can enable prevention, while retroactive identification of leaks can help an organisation repair the confidentiality of its systems and prevent recurring damage as well as determine impact by ascertaining specifics of loss.

Exfiltration Detection	Known Channel Inspection	Network Monitoring	Host Access Analysis/Logging	Leakage Flow Analysis (Post Hoc)	Digital Watermarking (Post Hoc)
Known Overt: HTTP, FTP, IM, P2P, Email/Webmail	✓	✓	✓	✓	✓
Unknown Overt: HTTP, FTP, IM, P2P, Email/Webmail	x	✓	%	%	✓
Encrypted: SSH, SCP, SFTP, HTTPS, VPN	%	%	✓	✓	✓
Protocol Tunnelling: e.g. HTTP, DNS, ICMP Tunnels	%	✓	✓	%	x
Steganography: Image, Network or VoIP Steganography	%	%	✓	✓	✓
Timing Channels	x	x	✓	✓	✓
Steganography + Encryption	x	x	✓	✓	✓

Table 1: The coverage of common approaches to exfiltration detection. A ✓ indicates this method usually detects exfiltration attempts of this type, a % indicates the method can be made to cover this threat, or partially covers this threat, and a × indicates this threat is not covered.

Of course, a definitive checklist is not possible or feasible given the diversity of security needs and security postures acceptable to different organisations. However, Table 1 provides a helpful guide to decide which solutions enable detection of particular data exfiltration methods. The table can be used to tailor an organisation's response to potential data exfiltration attempts. Known overt channels (e.g. HTTP, FTP, IM, P2P, Email/Webmail) are high-risk channels that are monitored through the deployment of filters at a channel-specific proxy or gateway (e.g. HTTP proxy or email proxy). For example, filters for these channels are configured as content-matching rules/signatures that allow files or protocol messages to be blocked on a positive match. Unknown overt channels, on the other hand, are overt channels which are not known to an organisation and therefore not monitored i.e. there are no filters in place to block data being exfiltrated via these channels. Encrypted channels used for exfiltration include SSH, SCP, SFTP, HTTPS or VPN. These use readily available encryption schemes such as SSL/TLS to obfuscate sensitive data and thwart the filters scanning traffic at the network boundary. Steganographic channels, which are growing in popularity for data exfiltration, include image, network and VoIP steganography. To further confuse content analysis attempts (for example, through the use of steganalysis filters), attackers may encrypt the data before embedding them into the steganographic medium.

Known Channel Inspection

One of the simpler approaches to detecting data exfiltration attempts is to inspect outgoing traffic on high-risk channels, searching for sensitive material through defined patterns, keywords or hashes. Proxy servers for each channel under inspection intercept outgoing data and submit it to content inspection products for analysis. Thus, proxy servers for email, IM, FTP channels enable the content of files to be handed to content analysers to look for matches in keywords, personally identifiable information, hashes, defined patterns or files flagged by signatures. A positive match identifies the data as sensitive and thus an unauthorised attempt to exfiltrate data across network boundaries. These tools cause transfers to be blocked by the channel's proxy server while alerting information security personnel.

A particularly common channel for this sort of monitoring is email. The ubiquity and simplicity of email as a transfer mechanism, combined with the relative ease with which it can be monitored via a mail proxy, make it a good target for detection systems. More advanced approaches look at identifying whether emails should be sent based on the topics a recipient is able to access[94;95], which has a dual benefit in that it prevents accidental leakage through misdirected mail.

Known channel systems are inherently limited to studying the content of the medium for which they are designed, and as such provide no protection against unexpected exfiltration channels. Mail filters, for example, cannot prevent HTTP uploads of confidential files. Naive approaches to the design of these systems would also fail to catch encrypted or steganographic files, but steganalysis modules for email filters are viable[81] and similar measures could be put in place to detect gibberish data in attachments and flag it as potentially encrypted material. Similar systems could be put in effect for other known channels. Timing channels are difficult to detect in any medium, and the favoured known channel for detection systems (email) would be an unlikely choice of vector for such a channel, so it is unsurprising that such a method is not countered.

Broadly speaking, known channel inspection methods such as mail filters can be effective at timely detection and prevention of overt exfiltration attempts via an anticipated means, but ultimately have poor coverage against the range of possible exfiltration threats and would be easily evaded by skilled attackers such as APTs.

Network Monitoring

An approach which broadens the coverage of threats while still providing timely detection is to monitor not a single channel but all network traffic moving out of an organisation. Specialised deep packet inspection (DPI) products can be used to inspect all outgoing data packets for overlaps with confidential data[96]. The impact of DPI on network performance can be troublesome for networks with large throughput, but there are systems designed to more rapidly handle such volumes, including hardware acceleration to minimise its overhead[76]. It is worth noting that DPI can be considered highly intrusive for legitimate users of a network, and that such intrusion may lead to increased purposeful evasion of the monitoring system. Attempts have been made to technologically balance the demands of privacy and enforcing confidentiality in designing such systems[40].

DPI systems should theoretically detect all undisguised transmission of sensitive materials, but encryption and steganography remain troublesome and indeed might be viewed as more of an issue than in analysis of a known channel, due to the wide range of possible non-malicious traffic. Again, some steganalysis tools exist [48;49] to help deal with protocol tunnelling and disguised traffic, but encryption remains a problem. Flagging all encrypted traffic passing through a network boundary as a possible leak would be an unhelpful approach, and yet it would also usually be undesirable – leaving aside questions of feasibility – to attempt decryption on purposefully secured traffic.

One solution to this is to monitor network packets not for their content, but to discover abnormal patterns of behaviour. Learning from an organisation's normal traffic patterns, tools can identify users[42], detect aberrations in traffic rates caused by exfiltration attempts[68] and flag suspicious data transfers even though the data being transmitted is encrypted[2]. Such an approach, appropriately deployed, can even be useful in detecting passive snooping on local wireless networks[34;75]. Abnormal behaviour detection is also useful to detect tunnelling. For instance, many of the DNS tunnelling tools available do not try to be stealthy; they rely on the fact that DNS is often not monitored. Analyses of DNS request/response payloads can help to detect tunnelling indicators. Similarly, the volume and frequency of traffic over a particular channel can be an indication of tunnelling.

Network anomaly detection using flow-based evaluation can help identify spikes in outbound traffic egressing border firewalls as exfiltration. A network flow is defined as a unidirectional sequence of packets that all share the same characteristics such as ingress interface, source/destination IP addresses, source/destination ports for TCP/UDP protocols, to name but a few. Network flow monitoring standards Netflow, sFlow or IPFIX and implementing tools can be used by administrators on the corporate network to determine the baseline 'normal' netflows between workstations and servers, and in particular network segments. This network-centric approach enables network administrators to monitor and control the flow of data within the network. Any anomalies that exceed the normal traffic patterns can thus be detected and appropriate action taken to investigate these outliers. Attackers often use an encrypted channel to bypass network security devices. An additional measure in monitoring all traffic leaving the organisation is to detect any unauthorised use of encryption (for example SSL/TLS connections with unknown IP addresses). This enables organisations to detect rogue connections, terminate them and investigate the infected source computer.

If an Intrusion Detection System (IDS) is in place, it is also recommended that DNS queries on unusual domain names (for example, outside the Access Control List) are blocked. This requires that DNS query logging is enabled to detect hostname lookup for known malicious C2 domains.

Host-based Access Analysis

An alternative to monitoring network traffic for already-outbound sensitive data is to monitor the storage system containing the data and making note of unusual patterns of access. This can be implemented as part of a database management system[12;41] for a central share, or even at the file system or system call level[11;20;63;35] for servers or employee machines.

Such systems need not be particularly complex in their modelling of suspicious activity and engage in fastidious monitoring activities that penalise performance. Simple logging and analysis of particular patterns of access to the file system, database or the OS's system call library can trigger alerts for exfiltration [40;25] – a sequence of operations such as ***encrypt* → *compress* → *http-web*** might, for example, be interpreted as an exfiltration attempt. Even non-remote threats are somewhat countered by such a system and, with specific countermeasures, host systems storing confidential data can not only detect but also protect themselves against duplication to unauthorised devices[39] or remote hosts[83]. Beyond the detection of individual events, such access analysis tools could prove useful in diagnosis of the techniques used by attackers[26], and in recovery efforts.

There are trade-offs between monitoring network traffic and monitoring host storage systems. For example, individually innocuous actions on many machines within an organisation might add up to a breach of proper access, which a host-based analysis system would not detect but a network-based monitor would consider suspicious[53]. Yet again, host-based access analysis could help protect data from being physically removed from an organisation's network on mobile devices[37;38;97] – a consideration apt for increasingly mobile modern work environments.

Leakage Flow Analysis

Technical detection of on-going exfiltration is one aspect of the problem. Organisations must also analyse the risk involved in trusting agents with information, and pay attention to identification of the components of a system which were involved in a data breach which has already been carried out. The value of such work lies in identification of attacker strategies[98], and these approaches have been effective in actual deployments[36].

Methods range from modelling the possible propagation of data [99] and risk involved in its dissemination[100] to more proactive methods whereby slightly different information is allocated to parties so that if leaks happen the culprit can be identified based on which version of the information was revealed [45;61].

Watermarking

A technique that appears in tandem with a number of other detection strategies is the digital watermarking of sensitive files. The relatively simple measure of a small signature inserted into documents for identification purposes or applied to relational databases[67] can help with post-hoc leakage point identification[51;72] by uniquely marking the origin of exfiltrated material and with real-time detection of exfiltration attempts by containing signatures to which a DPI product can remain alert[84].

The vulnerability of watermarking is its detection. If an attacker identifies and removes a digital watermark, the aid it offers to monitoring services as well as leakage tracing is removed. For this reason, watermarks need to be discreet when embedded in sensitive data or documents.

Case Study: IP Data Exfiltration

The purpose IP data exfiltration attacks includes industrial espionage as well as collecting intellectual property for competitive advantage. Such attacks may span over several months as the attackers perform reconnaissance of the network environment for further attack planning. The attackers seek to steal intellectual property such as design documents, source code, chemical formulae and manufacturing processes for advanced materials.

An example of such an attack is the Nitro attack, codenamed by Symantec, that targeted a total of 29 Fortune 500 companies in the chemical sector and another 19 in various other sectors, primarily the defence sector. The geographical distribution of the Nitro attacks meant that attackers targeted companies that they knew possessed some intellectual property of interest. An alternative reason may be because they believed these companies had less stringent security measures in place and therefore presented an easier access point for data exfiltration.

Anatomy of IP Data Exfiltration

IP data exfiltration attack prepetrators use similar tactics as in the previous case studies to gain access to the target organisation's network. The Nitro Attacks perpetrators, for instance, employed simple social engineering tactics to infiltrate the networks of the targeted organisations. They first researched desired targets and then sent an email specifically to each of them. Following common attackers' doctrines, they focused their attention on hacking the weakest link: the employees rather than attempting to defeat the infrastructure perimeter defences.

Whether it is through concealed executable files, weaponised documents or malicious links, the initial intrusion step results in the installation of malicious code (such as a rootkit) to enable initial host access and control. As it is often the case, an initial infection will result in the sequential download and installation of malicious software (for example, a remote access tool or a tool to capture password hashes) to enable a more effective control of the compromised machine and penetration of the network.

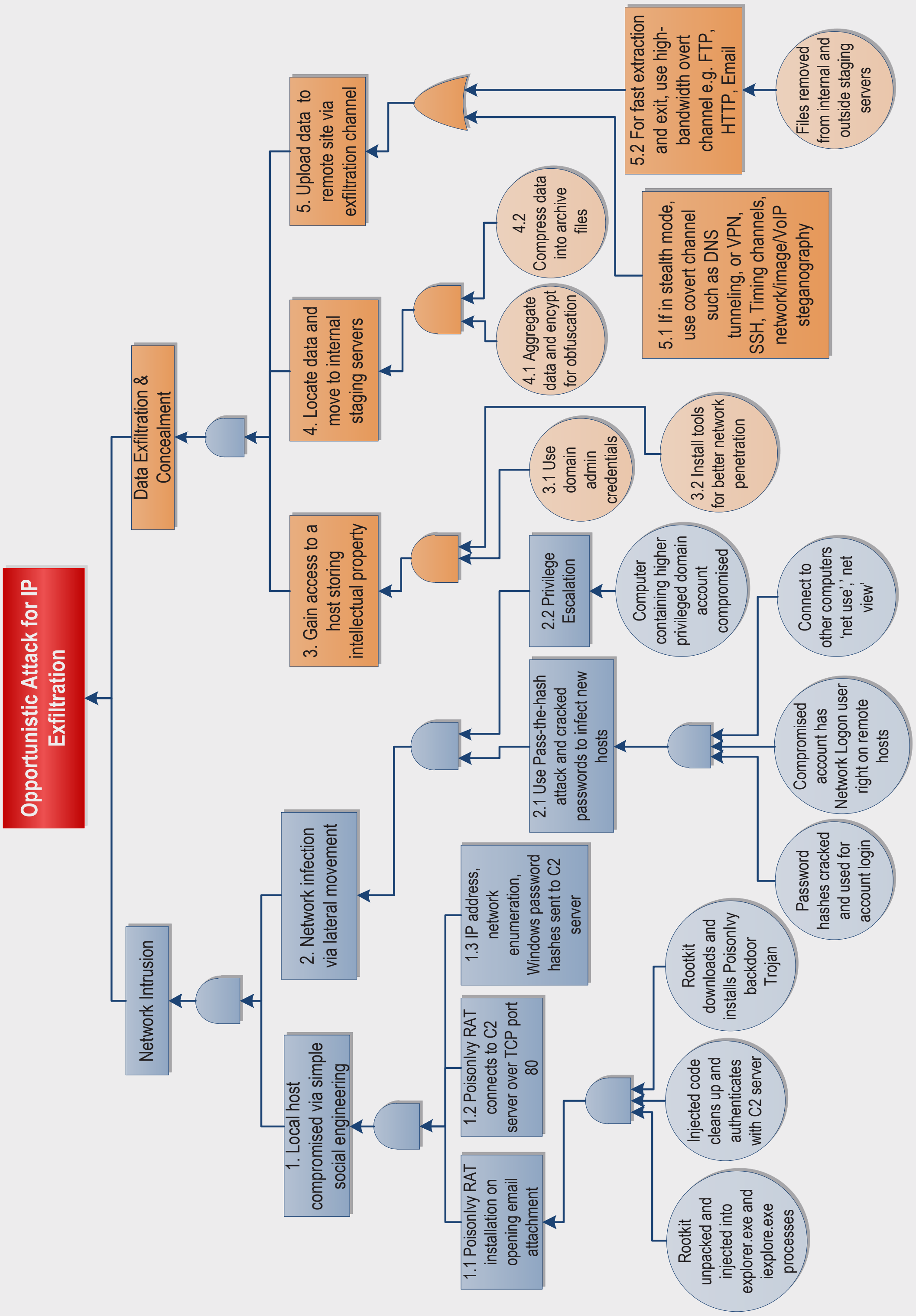
In the Nitro IP Data Exfiltration attacks, for instance, after the initial intrusion and rootkit deployment, the rootkit program, authenticated with the Command and Control server via TCP port 80 and upon success, received binary code containing Poison Ivy, a common backdoor Trojan.

The Poison Ivy Remote Access Tool (RAT), once downloaded and installed, provided remote access to the compromised host for external control of computers or servers. Usually RAT tools are configured to operate in a reverse-connect mode. In this mode, they pull commands from the central C2 servers and then execute the commands, rather than C2 servers initiating the command sequence. This type of connectivity enables them to avoid detection, as the traffic is mostly outbound from the compromised host to C2 rather than inbound. The communication between backdoors and their C2 servers (as is the case with the Poison Ivy Trojan) are often encrypted by the transmission protocol to obfuscate the messages from network monitoring tools that perform deep packet inspection.

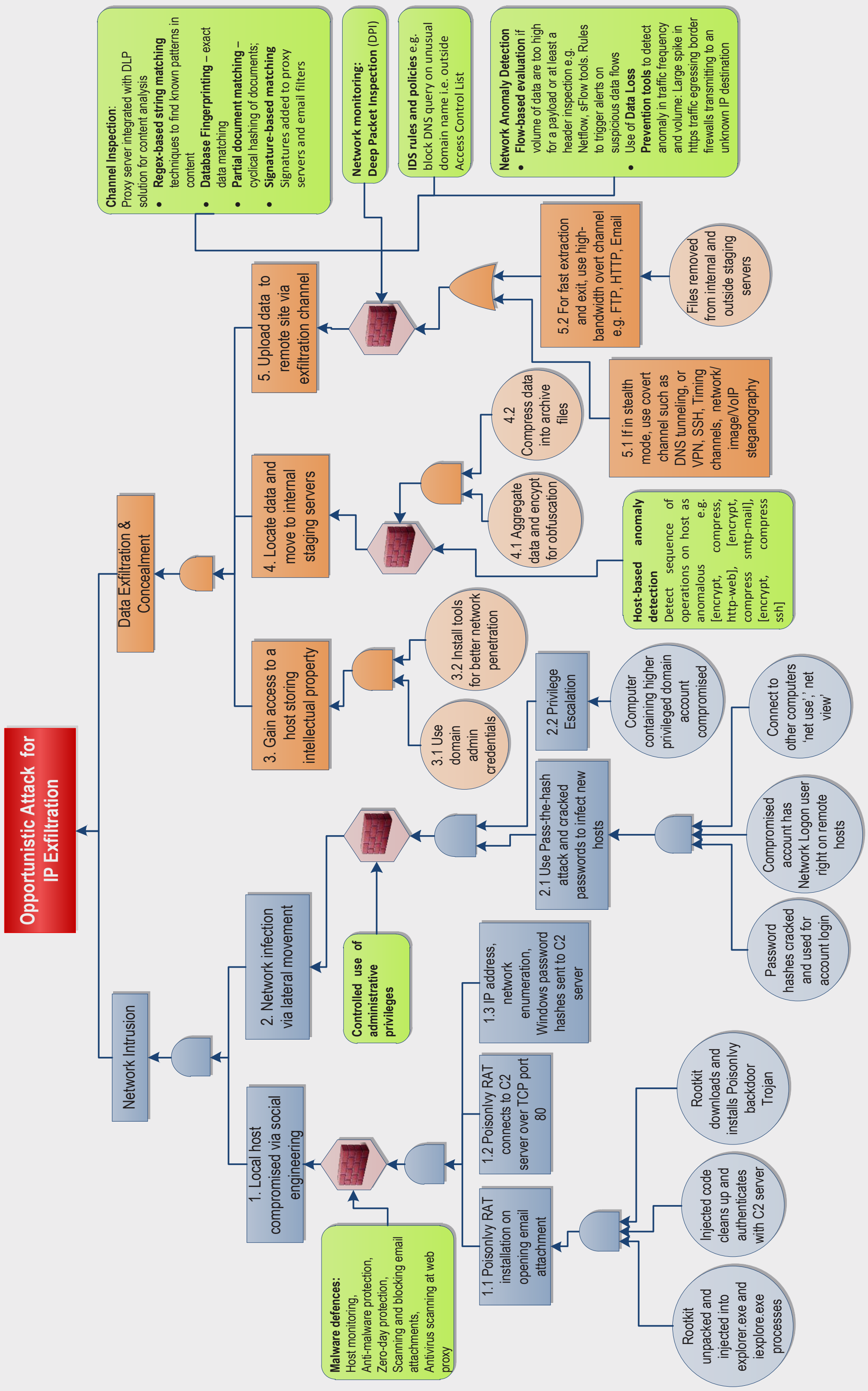
With the RAT in place and successfully authenticated, the attackers can then instruct the RAT instance to provide the infected computer's IP address, the names of all other computers in the workgroup or domain, and dumps of Windows cached password hashes. Using various tactics such as pass-the-hash attacks or cracked Windows password hashes, the attackers proceed to gain access to other computers with the same Network Logon user rights. In a typical attack scenario, an attacker after intrusion on a host may start digital shoulder-surfing to establish the employee's role and their level of access. If initial entry points may not be strategic enough for the attackers' purpose, they will seek user accounts with more elevated access rights to relevant services and servers. To this end, the attackers may perform privilege escalation on non-administrative users in the targeted systems, and then move on to gain access to key high value targets, which include process experts and IT and non-IT specific server administrators. With domain administrator credentials, for instance, it is easy for the attacker to traverse the network and find servers hosting the desired intellectual property and gain access to the sensitive materials.

Once the attackers have identified the desired intellectual property, they establish access to staging servers at key aggregation points on the network; this is done to prepare the data for exfiltration. Data is copied to internal staging servers, aggregated, compressed and encrypted for exfiltration. The exfiltration channel used depends on the degree of covertness of the attack, the volume of data to be exfiltrated and the likelihood that the exfiltration will be detected. For instance, some attackers maintain a presence in the enterprise networks for months waiting for valuable data. In such a case, using a covert exfiltration channel such as SSH or DNS tunnelling for transmitting data to external staging servers is less likely to be detected.

However, if the volume of data is large and the security controls in place make detection likely, the attackers' goal may be to steal as much valuable data as possible before detection. In this case, high-bandwidth overt channels such as FTP or HTTP are the usual recourse.


Opportunistic Attack for IP Exfiltration
Network Intrusion
Data Exfiltration & Concealment
1. Local host compromised via simple social engineering
2. Network infection via lateral movement
3. Gain access to a host storing intellectual property
4. Locate data and move to internal staging servers
5. Upload data to remote site via exfiltration channel
1.1 PoisonIvy RAT installation on opening email attachment
1.2 PoisonIvy RAT connects to C2 server over TCP port 80
1.3 IP address, network enumeration, Windows password hashes sent to C2 server
Rootkit unpacked and injected into explorer.exe and iexplore.exe processes
Injected code cleans up and authenticates with C2 server
Rootkit downloads and installs PoisonIvy backdoor Trojan
2.1 Use Pass-the-hash attack and cracked passwords to infect new hosts
2.2 Privilege Escalation
Password hashes cracked and used for account login
Compromised account has Network Logon user right on remote hosts
Connect to other computers 'net use',' net view'
Computer containing higher privileged domain account compromised
3.1 Use domain admin credentials
3.2 Install tools for better network penetration
4.1 Aggregate data and encypt for obfuscation
4.2 Compress data into archive files
5.1 If in stealth mode, use covert channel such as DNS tunneling, or VPN, SSH, Timing channels, network/image/VoIP steganography
5.2 For fast extraction and exit, use high-bandwidth overt channel e.g. FTP, HTTP, Email
Files removed from internal and outside staging servers

Anatomy of IP Data Exfiltration


Opportunistic Attack for IP Exfiltration
Network Intrusion
Data Exfiltration & Concealment
1. Local host compromised via social engineering
2. Network infection via lateral movement
3. Gain access to a host storing intellectual property
4. Locate data and move to internal staging servers
5. Upload data to remote site via exfiltration channel
1.1 Poisonlvy RAT installation on opening email attachment
1.2 Poisonlvy RAT connects to C2 server over TCP port 80
1.3 IP address, network enumeration, Windows password hashes sent to C2 server
Rootkit unpacked and injected into explorer.exe and iexplore.exe processes
Injected code cleans up and authenticates with C2 server
Rootkit downloads and installs Poisonlvy backdoor Trojan
2.1 Use Pass-the-hash attack and cracked passwords to infect new hosts
2.2 Privilege Escalation
Password hashes cracked and used for account login
Compromised account has Network Logon user right on remote hosts
Connect to other computers 'net use', net view'
Computer containing higher privileged domain account compromised
3.1 Use domain admin credentials
3.2 Install tools for better network penetration
4.1 Aggregate data and encypt for obfuscation
4.2 Compress data into archive files
5.1 If in stealth mode, use covert channel such as DNS tunneling, or VPN, SSH, Timing channels, network/ image/VoIP steganography
5.2 For fast extraction and exit, use high-bandwidth overt channel e.g. FTP, HTTP, Email
Files removed from internal and outside staging servers

Malware defences: Host monitoring, Anti-malware protection, Zero-day protection, Scanning and blocking email attachments, Antivirus scanning at web proxy

Controlled use of administrative privileges

Host-based anomaly detection Detect sequence of operations on host as anomalous e.g. [encrypt, compress, http-web], [encrypt, compress, smtp-mail], [encrypt, compress ssh]

Channel Inspection: Proxy server integrated with DLP solution for content analysis

- **Regex-based string matching** techniques to find known patterns in content
- **Database Fingerprinting** – exact data matching
- **Partial document matching** – cyclical hashing of documents;
- **Signature-based matching** Signatures added to proxy servers and email filters

Network monitoring: Deep Packet Inspection (DPI)

IDS rules and policies e.g. block DNS query on unusual domain name i.e. outside Access Control List

Network Anomaly Detection

- **Flow-based evaluation** if volume of data are too high for a payload or at least a header inspection e.g. Netflow, sFlow tools. Rules to trigger alerts on suspicious data flows
- Use of **Data Loss Prevention tools** to detect anomaly in traffic frequency and volume: Large spike in https traffic egressing border firewalls transmitting to an unknown IP destination

Countermeasures for detecting and preventing IP Data Exfiltration

Detecting and preventing IP Data Exfiltration

The data exfiltration in the Nitro Attacks concerned mainly data-at-rest i.e. data residing in file systems, databases and other storage methods. When applying data exfiltration security controls to data-at-rest, it is essential to identify the location of sensitive data in the organisation. This is performed using tools for data discovery scanning. Data discovery has two uses. The primary use of this control is for discovering sensitive data on data depositories. These tools include the capability to perform content inspection on documents and can, therefore, be configured with policies to look for sensitive content. Documents tagged as 'confidential' might be a simple example of such policies. Discovery scanning can also be used to generate string-comparison (regular-expressions) rules for content matching of files containing structured content. With regards to unstructured data, the scanner can generate fingerprinting data, hashes or file signatures that can be used by filters to detect exfiltration of this type of data. A common technique to protect documents is to tag document with particular keywords and then use an antivirus tool such as ClamAV to generate signatures (hexadecimal patterns). This signature can then be added to a proxy server (e.g., Squid) to block these documents from leaving the organisation, for example, as email attachments. A variant of this technique is the digital watermarking of sensitive files.

As can be seen in the figure opposite, possible barriers can be selected from the arsenal of exfiltration countermeasures and deployed. Examples include: channel inspection and network Monitoring (content inspection, deep packet inspection, signature-matching, IDS DNS rules, network anomaly detection), host-based access control and analysis, controlled use of administrative privileges, and malware defences.

Preventing and Mitigating Data Exfiltration

Prevention and mitigation systems are distinguished from detection systems in that while some systems for the detection of exfiltration can be used to prevent it, not all approaches to prevention will alert the owner that exfiltration has been attempted. As with detection measures, a definitive checklist of preventive measures is not possible or feasible given the varying needs and business contexts in which different organisations operate. However, Table 2 provides a helpful guide to decide which prevention and mitigation approaches are more suited to countering particular types of exfiltration methods. The table can be an aid to identifying suitable preventive measures as part of an organisation's information security and risk management strategy.

Exfiltration Prevention and Mitigation	Actuated Detection Systems	Security Policy Tools	Self-Protecting Data	Low-Level Snooping Defences
Known Overt: HTTP, FTP, IM, P2P, Email/Webmail	✓	✓	✓	✓
Unknown Overt: HTTP, FTP, IM, P2P, Email/Webmail	%	%	%	%
Encrypted: SSH, SCP, SFTP, HTTPS, VPN	%	✓	✓	✓
Protocol Tunnelling: e.g. HTTP, DNS, ICMP Tunnels	%	✓	✓	✓
Steganography: Image, Network or VoIP Steganography	%	✓	✓	✓
Timing Channels	x	✓	✓	✓
Steganography + Encryption	x	✓	✓	✓

Table 2: The coverage of common approaches to exfiltration prevention. A ✓ indicates this method usually prevents exfiltration attempts of this type, a % indicates the method can be made to cover this threat, or partially covers this threat, and a × indicates this threat is not covered.

Actuated Detection Systems

Many exfiltration detection systems can be empowered to automatically or semi-automatically block transfers they consider suspicious. This can be implemented on many levels, from email filters, which refuse to forward suspicious attachments[81] to database management systems that refuse to respond to suspicious queries[12;29], to deep packet inspection gateways that refuse to forward packets containing sensitive data[96].

The opportunities for actuation can be more fine-grained than simply allowing or denying access. In some cases it is beneficial to instead mask the sensitive portions of outgoing material[64;87], leaving non-sensitive information intact while protecting the organisation's confidentiality. More cunningly, an intelligent detection system could replace the requested sensitive information with plausible decoy information, beginning a retaliatory disinformation campaign against an attacker[82].

The strength of protection provided by an actuated detection system depends for the most part on the coverage of detectable threats, particularly the false negative rate – the rate at which the detection system misses exfiltration attempts. Systems which deny access in response to detection run the risk of alerting attackers that their attempts are being monitored, perhaps prompting them to make use of more covert methods, while systems which merely mask sensitive information may still reveal contextual information which aids attackers in recovering it.

Security Policy Assistance Tools

One of the most fundamental measures for mitigating exfiltration threat is properly defining security policies regarding granting and maintaining the access to and storage of sensitive material. While this countermeasure is primarily organisational, technological solutions exist to help with adherence to policy frameworks.

These solutions range from policy enforcement software which directly interpret formulations of security policy to protect local access to data [21;50;77] to systems which regulate the recipients of forwarded information according to policy[103], to systems for ensuring that third parties are adhering to your security policies [47;101].

Encryption is key in many security policies regarding interaction with cloud services, and related work looks specifically at how cloud services can provide assurances regarding their own encryption practices[102]. Perhaps the best systems, where possible, involve the encryption of data before it is sent to a cloud storage device[55], but this raises questions about service providers' ability to assure the integrity of encrypted data[102].

Self-Protecting Data

Many exfiltration prevention systems designed for static corporate networks suffer in mobile contexts, whether the mobility is physical (as with the increasing use of mobile devices which need access to sensitive data) or digital (as with data moved to virtualised storage space owned by a third party). A key modern requirement is for prevention of data theft while data is being duplicated from device to device in order to be accessed.

The obvious approach to such a requirement is to make use of encryption, storing data in a format which is unusable without the proper credentials. A number of encryption schemes are suitable for such a purpose[8] and encryption can easily be applied in an automatic manner to save effort on the user's part[32;43;46]. From such components, we reach the concept of self-protecting or security-aware data, where a software layer implementing corporate security policy wraps an encrypted container for sensitive data[73;77;78].

Such containers can be implemented in software[79;104] or with additional hardware support to assure integrity[12]. Self-protecting data schemes such as this rely on encryption to protect information from being stolen from storage devices, but could still be vulnerable to attacks stealing information from memory analysis on a compromised host, as well as more mundane security issues regarding the distribution and memorisation of security tokens to unlock the containers.

Low-level Snooping Defences

Due to the rising use of virtual machines for handling sensitive data, modern threat models must include the possibility that a malicious agent already has access to the physical machine, which is running your critical software. As a result, a number of defences against memory analysis and address bus leakage can be deployed.

General measures exist for the isolation of virtual machines by restricting access to addresses in the virtual machine's hypervisor[19;21], some even going so far as to restrict any interaction between co-hosted virtual machines[33]. Specific measures countering memory analysis make use of either rapid erasure of sensitive data from memory[24] or timely encryption of said data[15;80;85]. Some defences even include an operating system designed to work against leakage threats[21] while others require specific hardware solutions to protect against leakage[105].

Many of these measures aim to prevent data exfiltration by removing or hiding it from potential attackers' view, and commonly rely on the service provider to implement them. While the threat of a malicious hypervisor[27] remains potent, implementation of measures like these could be considered the sign of a trustworthy provider.

Miscellaneous

A number of other novel measures exist, which could be deployed against niche exfiltration threats. Examples include:

- In order to prevent unnecessary local access to data, 'mobile agents' can be used which perform queries on the behalf of a user[106].
- Public information can be combined with private information in a manner which protects said private information[71]. Alternatively, a negative image of the actual data may be stored, so attackers would appear only to steal non-information. [62]
- Passive sniffing and use of covert channels may be countered through encrypted one-time network addresses[86].

Future Trends

New and emerging technologies are shaping work practices and organisational cultures and, hence, the security landscape for the next 5-7 years. It is important that strategies, technologies, tools and techniques for detecting and preventing data exfiltration take into account the potential exfiltration channels enabled by these emerging trends. Our analysis has derived insights into these issues from technology analysis reports, news and Internet sources, via responses from a survey of security professionals on the expected future technologies and work practices in their organisations and through several open-ended interviews with (non-security) professionals on trends expected to impact the future workplace.

Of course, most technological trends are likely to have an impact on an organisation's security posture with regards to data exfiltration. However, there are three key trends that have the potential both in isolation and in combination to pose significant challenges in this regard.

Hyper-mobility

The prevalence of smart phones, tablets and applications and the resulting 'bring your own device' culture leads to mobility in a range of dimensions:

Mobility of devices, that is *physical mobility of work devices* (e.g., the use of laptops, tablets, smart phones instead of office desktop computers);

Mobility of device function, for instance, using personal devices both for work and entertainment and use of multiple devices for a given work, e.g., using a tablet and a smartphone to edit a given report;

Mobility of ownership, for example, using personal devices for work, selling off a device and getting a replacement at an individual's will;

Mobility of Work Location, whereby work is carried out from varying physical locations, such as home office, clients' site, coffee shops, libraries, hot-desk rental spaces, etc.

Exfiltration vectors

A number of exfiltration vectors are unleashed by such hyper-mobility. For example, use of a mobile device for multiple functions (such as entertainment, shopping, social networking, as well as work) implies access and use of less trustworthy sites, applications, and services. This, coupled with the access and storage of confidential data on the same device leads to obvious exfiltration threats. More so, the transfer of data between multiple devices, the use of various communication protocols by applications on such devices and the use of virtualised infrastructures opens up a range of possible exfiltration channels.

However, exfiltration countermeasures need to adapt to this hyper-mobile context as restricting mobility can be highly counter-productive, leading to loss of revenues, if it is not complemented by training and investment in workplace-provided and authenticated devices. In particular, use of own devices for work is widely accepted by employees primarily because they benefit from the convenience of this arrangement: they can work where and when they need without loss of any familiar or necessary applications, configurations and customisations. In fact, mobility restrictions are likely to foster additional exfiltration vectors through concealed use of personal mobile devices and concealed use of Internet (social media, games, etc.) via tunnelling or peer-to-peer protocols. In the former case, sensitive data can be exfiltrated through a compromised device, while the latter could render countermeasures such as known channel inspection and network monitoring ineffective.

Hyper-connectivity

Continuous connectivity is already perceived to be an essential utility for most businesses. When working from a variety of physical locations using a mobile device, users rely on various (often untrusted) wireless networks. Use of (untrusted) Wi-Fi networks is motivated by their cost-free availability in many public places (such as cafes and libraries) and convenience for working on the move.

Exfiltration Vectors

Any data transmitted via such third-party networks would be susceptible to eavesdropping. The ready availability and use of VoIP services, instant message applications and email and the like respectively open up a range of exfiltration channels such as VOIP steganography or exfiltration via encrypted attachments. Furthermore, the potential for wearable computers and their interactions with mobile devices and the wider Internet of Things that form part of an individual's personal network could be compromised to access and extract sensitive data. For instance, such data could be posted (following encryption) to a public site where such wearable devices post information which is later accessed by the attacker.

Mitigation of threats arising from such hyper-connectivity require effective support for employees while on the move and using third-party networks. This includes use of secure access protocols to the company's network and end-to-end encryption of all transmitted data. Furthermore, the scope of exfiltration monitoring tools should be extended incorporating close and deep analysis of network traffic to/from users utilising third-party networks.

Hyper-virtualisation

Cloud computing and Internet-based services are leading to an extreme form of virtualisation. Such virtualisation has three key dimensions:

Service virtualisation refers to some business providing or consuming services purely digitally, without any own physical resources. For instance, a company may be providing a service to a customer where, in actual fact, that business, in turn, sources the said service from another company. A company may be dependent on a number of other companies for provision or use of such virtual services. For instance, in addition to sourcing web hosting, a company may use a virtual accountancy service, whereby all its accountancy is carried out by a 3rd party via invoices passed to a Dropbox account. Service virtualisation is motivated by cost saving, for example, due to specialisation, as for web hosting service providers; savings on physical expenses, as in case of printing and postage costs for invoices; as well as efficient use of time and convenience.

Virtualisation of teams occurs due to hyper-mobility of employees who are assigned to work on the same task from their current (geographically far removed) locations. This can be motivated by cost-cutting considerations (e.g., the workforce in developing countries will be less expensive), but also – more significantly – by the desire to access highly skilled and specialised professionals.

Virtualisation of employment refers to use of digital only information exchange and contacts in making an employment offer to an individual. This practice is already used by many companies, and could include, for instance, (i) identification of potential employees through blog trawling or publications relevance (e.g., companies inviting postgraduates or researchers for interviews based on their publications relevant to the company's R & D), (ii) holding interviews with the candidates via teleconferencing tools (e.g., Skype, Google Hangout, and shared documents for real-time typing of solutions to posed problems).

Exfiltration Vectors

Several data exfiltration threats arise due to service virtualisation such as data storage and management processes and procedures at the service provider site. Countermeasures are often hard to implement given the lack of ownership of the virtualised service and infrastructure. However, data exfiltration prevention strategies should seek to establish whether: data is stored securely in encrypted format, access to the data is properly authorised, data is transmitted via secure communication protocols, and hardware used is up–to-date and secure.

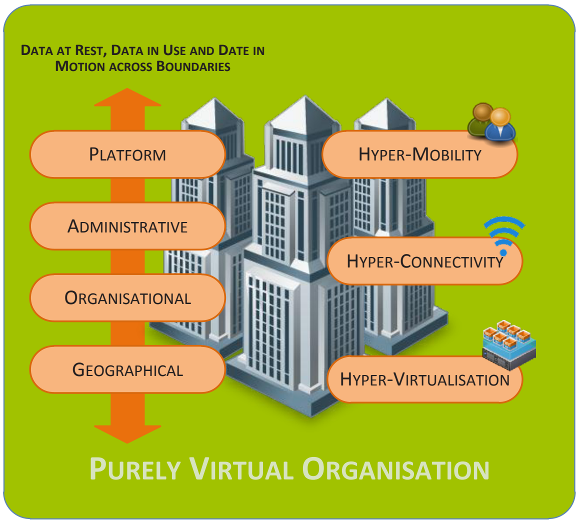
Furthermore, since virtual teams are fully dependent on telecommunications for their work, they often use such tools as instant messaging, email, file sharing, VoIP and so on – all of which substantially increase movement of data between devices, through (partially- and un-trusted) networks and a wide range of utility applications. All this could cause a continuous flux in software and hardware toolset that needs to be secured, monitored, and maintained by the virtual organisation, thus adding a number of potential exfiltration vectors (at least one per each data communication tool/technology and device) and stretching an organisation's ability to detect and prevent data exfiltration effectively.

Virtualisation of employment opens up advanced social engineering threats. An intruder could either deliberately set up blogs and postings that masquerade him/her as a suitable candidate for employment at a target company, or hijack another individual's identity, passing the interviews and thus, gaining access to data of interest to him/her.

Summary

The Rise of the Purely Virtual Organisation

Together, hyper-mobility, hyper-connectivity and hyper-virtualisation lead to the emergence of large purely virtual organisations fully dependent on service procurement and provision via the Internet and the cloud. Such an organisation does not own any offices – employees work from a wide variety of locations. It does not own any hardware – employees use their own hardware for business use or cloud-based services. Neither work force nor customers operate within any set physical or geographical boundaries. Such organisations already exist on various scales and are growing in number. They also participate in complex supply chains with other more traditional organisations (or those that are not yet at the extreme end of this hyper-mobile, hyper-connected, hyper-virtualised setting). As such the exfiltration vectors multiply to pose a variety of challenges. These arise from data crossing platform, administrative, organisational and geographical boundaries on a regular basis and requiring very sophisticated, yet unavailable, countermeasures to mitigate the threats posed by APTs and other attackers interesting in compromising data assets. Protecting data that is, for all intents and purposes, constantly in motion and in use across such an array of boundaries may yet be the biggest challenge we face with regards to security.



This document has covered typical data exfiltration means, counter measures and their effectiveness as well as key trends and patterns that may be indicative of data exfiltration from an organisation. We have also highlighted links between particular business practices and technologies and data exfiltration and how new and emerging technologies and/or business practices may impact data exfiltration modes, patterns and countermeasures.

The document is aimed at providing guidance that enables organisations to benchmark themselves against the state-of-the-art and state-of-the-practice in dealing with data exfiltration threats. Such a comparison is expected to reveal the areas where a particular entity (i.e., organisation) needs to improve its knowledge, expertise, competencies and technological innovation to either identify and thwart or recover with minimum loss from a data exfiltration attempt.

Our analysis of systems and approaches for detecting and preventing data exfiltration identifies three key areas in which a collective effort is required to improve security capability:

1. A stronger focus is needed on recovery post-exfiltration. Most approaches focus on detection, prevention and mitigation. However, there is no perfect solution to securing the increasingly complex corporate environments. As such recovery has to be a key focus of strategies, tools and techniques dealing with data exfiltration threats.
2. Current approaches and commercial systems mainly focus on policy specification and implementation for preventive measures. Examples of such measures include "drop all encrypted channels" as they cannot be examined by IDS or "block Dropbox access from sections of network holding sensitive data". However, such measures – though they may provide short-term solutions – do not fit in with modern working practices and often encourage users to find workarounds which, in itself, leads to further data exfiltration risks.
3. Given the complexity of modern organisational eco-systems and the emerging hyper- mobility, connectivity and virtualisation, the focus need to shift from 'Information Protection' only to an inclusive drive towards 'Detection and Prevention of Data Exfiltration'. This represents a more comprehensive philosophy that can cater for the threats and countermeasures needed for data at rest, in use and in motion.

Glossary

AD	Active Directory
APT	Advanced Persistent Threat
C2	Command and Control
DLP	Data Loss Prevention
DNS	Domain Name Service
DPI	Deep Packet Inspection
FTP	File Transfer Protocol
HICCUPS	Hidden Communication System for Corrupted Networks
HTTP	HyperText Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDS	Intrusion Detection System
IM	Internet Messaging
IP	Internet Protocol
IPS	Intrusion Prevention System
IRC	Internet Relay Chat
P2P	Peer to Peer
POS	Point of Sale
R & D	Research and Development
RAM	Random Access Memory
RAT	Remote Access Tool
RDP	Remote Desktop Protocol
RTCP	RTP Control Protocol
RTP	Real-time Transport Protocol
SCP	Secure Copy
SFTP	SSH File Transfer Protocol
SIP	Session Initiation Protocol
SQLi	SQL Injection
SSH	Secure Shell
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
VLAN	Virtual Local Area Network
VNC	Virtual Network Computing
VoIP	Voice over IP

References

1 Open Security Foundation, DataLossDB, data exfiltration repository. http://datalossdb.org/

2 Verizon, "2013 Data Breach Investigation Report", 2013.

3 Mandiant, "M-Trends – Attack the Security Gap", 2013 Threat Report.

4 PWC, "Information Security Breaches Survey 2010", Technical Report. InfoSecurity Europe, 2011.

5 IBM, "IBM Security Services Cyber Security Intelligence Index: Analysis of cyber security attack and incident data from IBM's worldwide security operations", IBM Global Technology Services, 2013.

6 Joe Pletcher Hannah Pruse Masoud Valafar Kevin Butler Adam Bates, Benjamin Mood. On detecting co-resident cloud instances using network flow watermarking techniques. *International Journal of Information Security, 2013.*

7 G. Al-Bataineh, A.; White. Analysis and detection of malicious data exfiltration in web traffic. *Malicious and Unwanted Software (MALWARE), 2012 7th International Conference on,* 2012.

8 K.; Ramli A.R. Alomari, M.A.; Samsudin. A study on encryption algorithms and modes for disk encryption. *2009 International Conference on Signal Processing Systems*, 2009.

9 Jason Andress and Steve Winterfeld. *Chapter 6 - Logical Weapons*. 2014.

10 Adam Bates, Benjamin Mood, Joe Pletcher, Hannah Pruse, Masoud Valafar, and Kevin Butler. Detecting co-residency with active traffic analysis techniques. In *Proceedings of the 2012 ACM Workshop on Cloud Computing Security Workshop,* 2012.

11 H.; Sutherland I. Benham, A.; Read. Network attack analysis and the behaviour engine. *Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on,* 2013.

12 Elisa Bertino and Gabriel Ghinita. Towards mechanisms for detection and prevention of data exfiltration by insiders. *Proceedings of the 6th International Symposium on Information, Computer and Communications Security, ASIACCS 2011,* 2011.

13 Jorge Blasco, Julio Cesar Hernandez-Castro, Juan E. Tapiador, and Arturo Ribagorda. Bypassing information leakage protection with trusted applications. *Computers & Security,* 2012.

14 Matthew Burnside and Angelos D. Keromytis. F3ildCrypt: End-to-End Protection of Sensitive Information in Web Services. In *INFORMATION SECURITY, PROCEEDINGS,* 2009.

15 Mustafa Canim, Murat Kantarcioglu, Bijit Hore, and Sharad Mehrotra. Building disclosure risk aware query optimizers for relational databases. *Proc. VLDB Endow.,* 2010.

16 Gary Cantrell and David D. Dampier. Experiments in hiding data inside the file structure of common office documents: a stegonography application. In *Proceedings of the 2004 International Symposium on Information and Communication Technologies,* 2004.

17 Yu-Yuan Chen, Pramod A. Jamkhedkar, and Ruby B. Lee. A software-hardware architecture for self-protecting data. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security,* 2012.

18 Justin Clarke. *Chapter 4 - Exploiting SQL Injection.* 2009.

19 D.; Hamama A.; Har'el N.; Kolodner E.K.; Kurmus A.; Shulman-Peleg A.; Sorniotti A. Factor, M.; Hadas. Secure logical isolation for multi-tenancy in cloud storage. *Mass Storage Systems and Technologies (MSST), 2013 IEEE 29th Symposium on,* 2013.

20 A. Flood, J.; Keane. A proposed framework for the active detection of security vulnerabilities in multi-tenancy cloud systems. *Emerging Intelligent Data and Web Technologies (EIDWT), 2012 Third International Conference on,* 2012.

21 Hanjun Gao, Lina Wang, Wei Liu, Yang Peng, and Hao Zhang. Preventing secret data leakage from foreign mappings in virtual machines. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering,* 2012.

22 Annarita Giani, Vincent H. Berk, and George V. Cybenko. Data exfiltration and covert channels. *Proceedings of SPIE - The International Society for Optical Engineering,* 2006.

23 A.S.; Beyah R.; Copeland J.A. Goldman, A.D.; Uluagac. Plugging the leaks without unplugging your network in the midst of disaster. *Local Computer Networks (LCN), 2012 IEEE 37th Conference on,* 2012.

24 Kalpana Gondi, Prithvi Bisht, Praveen Venkatachari, A. Prasad Sistla, and V. N. Venkatakrishnan. Swipe: eager erasure of sensitive data in large scale systems software. In *Proceedings of the Second ACM Conference on Data and Application Security and Privacy,* 2012.

25 Jonathan Grier. Detecting data theft using stochastic forensics. *Digital Investigation,* 2011.

26 R. Hassan, S.; Guha. Security and integrity analysis using indicators. *Cyber Security (CyberSecurity), 2012 International Conference on,* 2012.

27 K. Hay, B.; Nance. Circumventing cryptography in virtualized environments. *Malicious and Unwanted Software (MALWARE), 2012 7th International Conference on,* 2012.

28 Mario Heiderich, Marcus Niemietz, Felix Schuster, Thorsten Holz, and J Schwenk. Scriptless attacks: stealing the pie without touching the sill. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security,* 2012.

29 Li Shan; Dong Xiaorui; Rao Hong. An adaptive method preventing database from sql injection attacks. *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on,* 2010.

References (continued)

30 Lin-Shung Huang, Zack Weinberg, Chris Evans, and Collin Jackson. Protecting browsers from cross-origin css attacks. In *Proceedings of the 17th ACM Conference on Computer and Communications Security,* 2010.

31 William Huba, Bo Yuan, Daryl Johnson, and Peter Lutz. A http cookie covert channel. In *Proceedings of the 4th International Conference on Security of Information and Networks,* 2011.

32 K. Izumi, M.; Horikawa. Toward practical use of virtual smartphone. *Information and Telecommunication Technologies (APSITT), 2012 9th Asia-Pacific Symposium on,* 2012.

33 Trent Jaeger, Reiner Sailer, and Yogesh Sreenivasan. Managing the risk of covert information flows in virtual machine systems. In *Proceedings of the 12th ACM Symposium on Access Control Models and Technologies,* 2007.

34 M.; Dijiang Huang Jagasia. Distributed data-theft detection in wireless sensor networks. *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE,* 2009.

35 Brian Jewell and Justin Beaver. Host-Based Data Exfiltration Detection via System Call Sequences. In *PROCEEDINGS OF THE 6TH INTERNATIONAL CONFERENCE ON INFORMATION WARFARE AND SECURITY,* 2011.

36 W.; Bailey M.; Pal P.; Jahanian F.; Sanders-W.H. Jing Zhang; Berthier, R.; Rhee. Safeguarding academic accounts and resources with the university credential abuse auditing system. *Dependable Systems and Networks (DSN), 2012 42nd Annual IEEE/IFIP International Conference on,* 2012.

37 A.; Voas J. Johnson, R.; Zhaohui Wang; Stavrou. Exposing software security and availability risks for commercial mobile devices. *Reliability and Maintainability Symposium (RAMS), 2013 Proceedings - Annual,* 2013.

38 Neil F. Johnson, Phil A. Sallee, and John G. Voeller. *Detection of Hidden Information, Covert Channels and Information Flows*. 2008.

39 Theodoros Kavallaris and Vasilios Katos. On the detection of pod slurping attacks. *Computers & Security,* 2010.

40 Jinhyung Kim and Hyung Jong Kim. A study on privacy preserving data leakage prevention system. *Lecture Notes in Electrical Engineering,* 2012.

41 Seung Kim, Nam Wook Cho, Young Joo Lee, Suk-Ho Kang, Taewan Kim, Hyeseon Hwang, and Dongseop Mun. Application of density-based outlier detection to database activity monitoring. *INFORMATION SYSTEMS FRONTIERS,* 2013.

42 R. Koch. Towards next-generation intrusion detection. *Cyber Conflict (ICCC), 2011 3rd International Conference on,* 2011.

43 A Koyfman. Securing sensitive data with the Ingrian DataSecure Platform. In *FINANCIAL CRYPTOGRAPHY AND DATA SECURITY,* 2005.

44 K.Z.; Monrose F. Krishnan, S.; Snow. Trail of bytes: New techniques for supporting data provenance and limiting privacy breaches. *Information Forensics and Security, IEEE Transactions on,* 2012.

45 Ajay Kumar, Ankit Goyal, Ashwani Kumar, Navneet Kumar Chaudhary, and S. Sowmya Kamath. Comparative evaluation of algorithms for effective data leakage detection. *2013 IEEE Conference on Information and Communication Technologies, ICT 2013,* 2013.

46 Xiaosong Zhang; Fei Liu; Ting Chen; Hua Li. Research and application of the transparent data encprytion in intranet data leakage prevention. *Computational Intelligence and Security, 2009. CIS '09. International Conference on,* 2009.

47 Dan Lin and Anna Squicciarini. Data protection models for service provisioning in the cloud. In *Proceedings of the 15th ACM Symposium on Access Control Models and Technologies,* 2010.

48 Yali Liu, Cherita Corbett, Ken Chiang, Rennie Archibald, Biswanath Mukherjee, and Dipak Ghosal. Detecting sensitive data exfiltration by an insider attack. In *Proceedings of the 4th Annual Workshop on Cyber Security and Information Intelligence Research: Developing Strategies to Meet the Cyber Security and Information Intelligence Challenges Ahead,* 2008.

49 Yali Liu, Cherita Corbett, Ken Chiang, Rennie Archibald, Biswanath Mukherjee, and Dipak Ghosal. Sidd: A framework for detecting sensitive data exfiltration by an insider attack. *Proceedings of the 42nd Annual Hawaii International Conference on System Sciences, HICSS,* 2009.

50 Jun Ma, Zhiying Wang, Jiangchun Ren, Jiangjiang Wu, Yong Cheng, and Songzhu Mei. The application of chinese wall policy in data leakage prevention. *Proceedings - International Conference on Communication Systems and Network Technologies, CSNT 2012,* 2012.

51 Janusz Marecki, Mudhakar Srivatsa, and Pradeep Varakantham. A decision theoretic approach to data leakage prevention. *Proceedings - SocialCom 2010: 2nd IEEE International Conference on Social Computing, PASSAT 2010: 2nd IEEE International Conference on Privacy, Security, Risk and Trust,* 2010.

52 Steve Moyle. The blackhat's toolbox: SQL injections. *Network Security,* 2007.

53 Justin Myers, Michael Grimaila, and Robert Mills. Insider Threat Detection Using Distributed Event Correlation of Web Server Logs. In *PROCEEDINGS OF THE 5TH INTERNATIONAL CONFERENCE ON INFORMATION WARFARE AND SECURITY,* 2010.

54 Robert C. Newman. Covert computer and network communications. In *Proceedings of the 4th Annual Conference on Information Security Curriculum Development,* 2007.

55 R.K.; Lakshmi R.S. Nirmala, V.; Sivanandhan. Data confidentiality and integrity verification using user authenticator scheme in cloud. *Green High Performance Computing (ICGHPC), 2013 IEEE International Conference on,* 2013.

56 <no author>. *Chapter 1 - Introduction to Instant Messaging.* 2005.

57 <no author>. *Chapter 2 - AOL Instant Messenger (AIM).* 2005.

58 <no author>. *Chapter 3 - Yahoo! Messenger.* 2005.

59 <no author>. *Chapter 4 - MSN Messenger.* 2005.

60 <no author>. *Chapter 6 - Trillian, Goodgle Talk, and Web-based Clients.* 2005.

61 Panagiotis Papadimitriou and Hector Garcia-Molina. A model for data leakage detection. *Proceedings - International Conference on Data Engineering,* 2009.

62 N.; Eirinaki M. Patel, A.; Sharma. Negative database for data security. Computing, Engineering and Information, 2009. ICC '09. International Conference on, 2009.

63 U. Patel, P.C.; Singh. Detection of data theft using fuzzy inference system. *Advance Computing Conference (IACC), 2013 IEEE 3rd International,* 2013.

64 Sara Porat, Boaz Carmeli, Tamar Domany, Tal Drory, Ksenya Kveler, Alex Melament, and Haim Nelken. Masking Gateway for Enterprises. In *LANGUAGES: FROM FORMAL TO NATURAL,* 2009.

65 Mani Potnuru. LIMITS OF THE FEDERAL WIRETAP ACT'S ABILITY TO PROTECT AGAINST WI-FI SNIFFING. *MICHIGAN LAW REVIEW,* 2012.

66 Stacy Prowell, Rob Kraus, and Mike Borkin. *CHAPTER 4 - Protocol Tunneling.* 2010.

67 Jerry Kiernan Rakesh Agrawal, Peter J. Haas. Watermarking relational data: framework, algorithms and analysis. *The VLDB Journal,* 2003.

68 S.; Bidyarthy A.S. Ramachandran, R.; Neelakantan. Behavior model for detecting data exfiltration in network environment. *Internet Multimedia Systems Architecture and Application (IMSAA), 2011 IEEE 5th International Conference on,* 2011.

69 P. Ranjith, Chandran Priya, and Kaleeswaran Shalini. On covert channels between virtual machines. *Journal in Computer Virology,* 2012.

70 Alberto Revelli. *Chapter 4 - Exploiting SQL injection.* 2012.

71 Christophe Salperwyck, Nicolas Anciaux, Mehdi Benzine, Luc Bouganim, Philippe Pucheral, and Dennis Shasha. Ghostdb: hiding data from prying eyes. In *Proceedings of the 33rd International Conference on Very Large Data Bases,* 2007.

72 Rainer Schick and Christoph Ruland. Data leakage tracking - non-repudiation of forwarding. *Communications in Computer and Information Science,* 2011.

73 S.; Schwarzkopf R.; Freisleben B. Schmidt, M.; Fahl. Trustbox: A security architecture for preventing data breaches. *Parallel, Distributed and Network-Based Processing (PDP), 2011 19th Euromicro International Conference on,* 2011.

74 Paulo Shakarian, Jana Shakarian, and Andrew Ruef. Chapter 10 - How Iraqi Insurgents Watched U.S. Predator Video: *Information Theft on the Tactical Battlefield.* 2013.

75 M. Sigholm, J.; Raciti. Best-effort data leakage prevention in inter-organizational tactical manets. *MILITARY COMMUNICATIONS CONFERENCE, 2012 - MILCOM 2012,* 2012.

76 A. Smallwood, D.; Vance. Intrusion analysis with deep packet inspection: Increasing efficiency of packet based investigations. *Cloud and Service Computing (CSC), 2011 International Conference on,* 2011.

77 Anna Cinzia Squicciarini, Giuseppe Petracca, and Elisa Bertino. Adaptive data management for self-protecting objects in cloud computing systems. In *Proceedings of the 8th International Conference on Network and Service Management,* 2013.

78 Anna Cinzia Squicciarini, Giuseppe Petracca, and Elisa Bertino. Adaptive data protection in distributed systems. In *Proceedings of the Third ACM Conference on Data and Application Security and Privacy,* 2013.

79 S.; Lin D. Squicciarini, A.; Sundareswaran. Preventing information leakage from indexing in the cloud. *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on,* 2010.

80 P.; Iyer V.; Kanitkar A.; Sanjeev S.; Lodhia J. Srinivasan, R.; Dasgupta. A multi-factor approach to securing software on client computing platforms. *Social Computing (SocialCom), 2010 IEEE Second International Conference on,* 2010.

81 Veroniki Stamati-Koromina, Christos Ilioudis, Richard Overill, Christos K. Georgiadis, and Demosthenes Stamatis. Insider threats in corporate environments: A case study for data leakage prevention. *ACM International Conference Proceeding Series,* 2012.

82 Salvatore J. Stolfo, Malek Ben Salem, and Angelos D. Keromytis. Fog computing: Mitigating insider data theft attacks in the cloud. *Proceedings - IEEE CS Security and Privacy Workshops, SPW 2012,* 2012.

83 N.; Kumar R.; Thanudas B. Suresh, N.R.; Malhotra. An integrated data exfiltration monitoring tool for a large organization with highly confidential data source. *Computer Science and Electronic Engineering Conference (CEEC), 2012 4th,* 2012.

References (continued)

84 Yu Shyang Tan, Ryan K.L. Ko, Peter Jagadpramana, Chun Hui Suen, Markus Kirchberg, Teck Hooi Lim, Bu Sung Lee, Anurag Singla, Ken Mermoud, Doron Keller, and Ha Duc. Tracking of data leaving the cloud. *Proc. of the 11th IEEE Int. Conference on Trust, Security and Privacy in Computing and Communications, TrustCom-2012 - 11th IEEE Int. Conference on Ubiquitous Computing and Communications, IUCC-2012,* 2012.

85 Yang Tang, Phillip Ames, Sravan Bhamidipati, Ashish Bijlani, Roxana Geambasu, and Nikhil Sarda. Cleanos: Limiting mobile data exposure with idle eviction. In *Proceedings of the USENIX Conference on Operating Systems Design and Implementation, Berkeley, CA, USA,* 2012.

86 J. Trostle. Applying network address encryption to anonymity and preventing data exfiltration. *Military Communications Conference, 2008. MILCOM 2008. IEEE,* 2008.

87 Hiroshi Tsuda, Akihiko Matsuo, Kenichi Abiru, and Takayuki Hasebe. Inter-Cloud Data Security for Secure Cloud-Based Business Collaborations. *FUJITSU SCIENTIFIC & TECHNICAL JOURNAL,* 2012.

88 Adam L. Young and Moti M. Yung. On fundamental limitations of proving data theft. *IEEE Transactions on Information Forensics and Security,* 2006.

89 K Szczypiorski. HICCUPS: Hidden communication system for corrupted networks. In *Proceedings of: The Tenth International Multi-Conference on Advanced Computer Systems ACS'2003, October 22-24, 2003* Międzyzdroje

90 Skoudis, E. (2012, February 29). The six most dangerous new attack techniques and what's coming next?. Retrieved from https://blogs.sans.org/pentesting/files/2012/03/RSA-2012-EXP-108-Skoudis-Ullrich.pdf

91 Van Leijenhorst, T. (2008). On the viability and performance of dns tunneling. Retrieved from http://www.uow.edu.au/~kwanwu/DNSTunnel.pdf

92 Dietrich, C. (2011, September 2). Feederbot - a bot using dns as carrier for its c&c. Retrieved from http://blog.cj2s.de/archives/28-Feederbot-a-bot-using-DNS-ascarrier-for-its-CC.html

93 Mullaney, C. (2011, August 31). Morto worm sets a (dns) record. Retrieved from http://www.symantec.com/connect/blogs/morto-worm-sets-dns-record

94 Polina Zilberman, Gilad Katz, Asaf Shabtai, and Yuval Elovici. Analyzing group e-mail exchange to detect data leakage. *Journal of the American Society for Information Science and Technology,* 2013.

95 S.; Katz G.; Elovici-Y.; Shabtai A. Zilberman, P.; Dolev. Analyzing group communication for preventing data leakage via email. *Intelligence and Security Informatics (ISI), 2011 IEEE International Conference on,* 2011.

96 Junchen Jiang; Yi Tang; Bin Liu; Yang Xu; Xiaofei Wang. Skip finite automaton: A content scanning engine to secure enterprise networks. *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE,* 2010.

97 F.C.C. Zhibo Zhao; Osono. 'trustdroid': Preventing the use of smartphones for information leaking in corporate networks through the used of static analysis taint tracking. *Malicious and Unwanted Software (MALWARE), 2012 7th International Conference on,* 2012.

98 Xiao Wang, Jinqiao Shi, and Li Guo. Towards analyzing traceability of data leakage by malicious insiders. *Communications in Computer and Information Science,* 2013.

99 Junfeng Yu, Shengzhi Zhang, Peng Liu, and ZhiTang Li. Leakprober: a framework for profiling sensitive data leakage paths. *In Proceedings of the First ACM Conference on Data and Application Security and Privacy,* 2011.

100 Zhengwei Yu; Yumei Wu. Risk assessment of customer information in telecommunication industry. *Information Science and Management Engineering (ISME), 2010 International Conference of,* 2010.

101 Rong-Wei Yu, Li-Na Wang, Xiao-Yan Ma, and Jin Ke. Flexible attestation of policy enforcement for sensitive dataflow leakage prevention. *1st International Conference on Multimedia Information Networking and Security, MINES 2009,* 2009.

102 ZeXing Hu Gail-Joon Ahn HongXin Hu Yan Zhui, HuaiXi Wang. Zero-knowledge proofs of retrievability. *Science China Information Sciences,* 2011.

103 Qihua Wang and Hongxia Jin. Data leakage mitigation for discretionary access control in collaboration clouds. *Proceedings of ACM Symposium on Access Control Models and Technologies, SACMAT,* 2011.

104 Jiangjiang Wu, Jie Zhou, Jun Ma, Songzhu Mei, and Jiangchun Ren. An active data leakage prevention model for insider threat. *Proceedings - 2011 International Symposium on Intelligence Information Processing and Trusted Computing, IPTC 2011,* 2011.

105 Xiaotong Zhuang, Tao Zhang, and Santosh Pande. Hide: an infrastructure for efficiently protecting information leakage on the address bus. *In Proceedings of the 11th International Conference on Architectural Support for Programming Languages and Operating Systems,* 2004.

106 Guido van t Noordende, Frances M. T. Brazier, and Andrew S. Tanenbaum. Guarding security sensitive content using confined mobile agents. *In Proceedings of the 2007 ACM Symposium on Applied Computing,* 2007.

107 Wojciech Mazurczyk, VoIP Steganography and Its Detection - A Survey. *In CoRR Journal, Volume abs/1203.4374,* 2012. http://arxiv.org/abs/1203.4374.

108 Squeeza SQL injection tool. Available at https://github.com/sensepost/squeeza

This material is provided for general information purposes only. You should make your own judgement as regards use of this material and seek independent professional advice on your particular circumstances. Neither the publisher, nor the author, nor any contributors assume any liability to anyone for any loss or damage caused by any error or omission in the work, whether such error or omission is the result of negligence or any other cause.

Security Lancaster
InfoLab21
Lancaster University
Lancaster LA1 4WA
United Kingdom

T: +44 (0)1524 510316
E: marash@comp.lancs.ac.uk

www.security-centre.lancs.ac.uk

www.fusiondesign.co.uk