

# Technical Aspects of Cyber Kill Chain

Tarun Yadav<sup>(✉)</sup> and Arvind Mallari Rao

Defence Research and Development Organisation, New Delhi, India  
{tarunyadav, arvindrao}@hqdrdo.in

**Abstract.** Recent trends in targeted cyber-attacks has increased the interest of research in the field of cyber security. Such attacks have massive disruptive effects on organizations, enterprises and governments. Cyber kill chain is a model to describe cyber-attacks so as to develop incident response and analysis capabilities. Cyber kill chain in simple terms is an attack chain, the path that an intruder takes to penetrate information systems over time to execute an attack on the target. This paper broadly categories the methodologies, techniques and tools involved in cyber-attacks. This paper intends to help a cyber security researcher to realize the options available to an attacker at every stage of a cyber-attack.

**Keywords:** Reconnaissance · RAT · Exploit · Cyber attack · Persistence · Command and control

## 1 Introduction

One of the leading problems faced by organizations is the emergence of targeted attacks conducted by adversaries who have easy access to sophisticated tools and technologies with an aim at establishing a persistent and undetected presence in the targeted cyber infrastructure. These multi-staged attacks are now becoming more complex, involving vertical and horizontal movement across multiple elements of the organization. The security research community has given this multi-stage chain of events culminating to cyber espionage a name: *The Cyber Kill Chain*. This paper aims on providing a primer on the cyber kill chain and surveys the recent trends and methodologies of the attacker at each stage of the cyber kill chain.

The paper talks about attacks in general irrespective of operating systems or application software since Cyber Kill Chain is a process rather than a technology. Technologies involved at each stage of the cyber kill chain process is explained without going into much details.

The paper is organized into 4 sections. Section 2 introduces the phases of cyber kill chain, Sect. 3 discusses the technical trends at each step of the cyber kill chain. The paper ends with concluding remarks in Sect. 4.

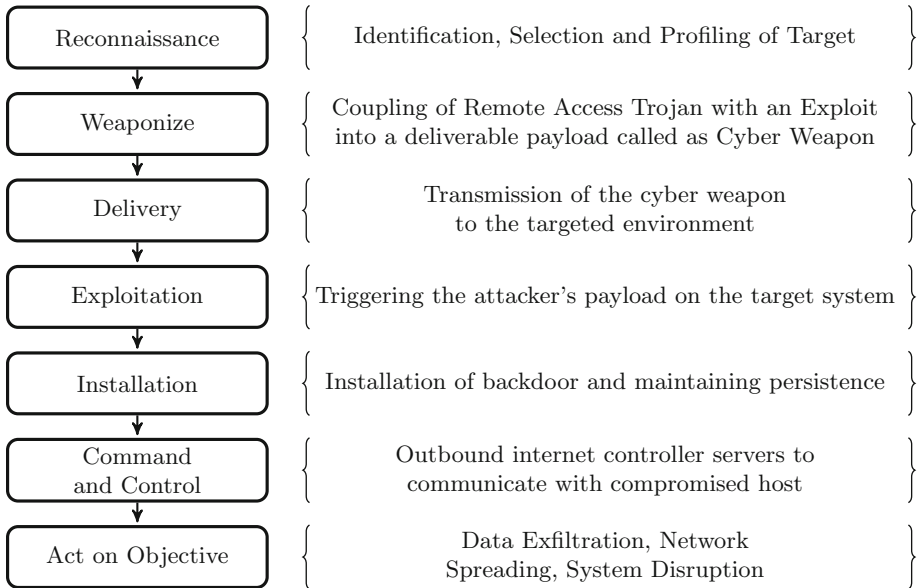
## 2 Cyber Kill Chain

Cyber kill chain is a model for incident response teams, digital forensic investigators and malware analysts to work in a chained manner. Inherently understanding, Cyber kill chain is modeling and analyzing offensive actions of a cyber-attacker [1]. So, for a security analyst who develops defensive counter measures [3], it is of utmost importance to study the cyber kill chain. This knowledge can help one think on the same lines of that of an attacker. Each phase of the kill chain in itself is a vast research area to tackle and analyze.

In recent years, cyber attacks have been more complex than it used to be and hence more destructive and dangerous [14, 41]. Nowadays multiple redundant attack vectors are being exploited in cyber attacks to not only multiply the effect but also making it more difficult for the response team to analyze.

To analyze such attacks, cyber kill chain provides a framework to breakdown a complicated attack into mutually nonexclusive stages or layers. Such a layered approach will enable the analysts to tackle smaller and easier problems and at the same time it will also help the defenders to subvert each phase by developing defenses and mitigation for each of the phases. Cyber Kill chain mainly consists of 7 phases [5, 6, 8] as shown in Fig. 1.

There are many articles [2, 4, 7] which describe the *Cyber Kill Chain* in detail with respect to recent attacks but most of them do not discuss the tools and technologies used by the attacker at each stage of the cyber kill chain. In the next



**Fig. 1.** Phases of cyber kill chain

section we will go through technical aspects involved at each step with respect to the attacker’s perspective and will see the variety of tools and methodologies utilized by the attacker.

### 3 Technical Aspects of Cyber Kill Chain

As described cyber kill chain defines the flow of a cyber attack and in this 7-layer model each layer is critical. Studying the cyber kill chain will help cyber threats to be identified or mitigated at any layer of attack. Sooner the detection is done lesser is the loss to the organization under attack. This section broadly outlines the technical methodologies, implementation, research, and tools involved at each stage of the cyber kill chain with examples of famous cyber attacks and malware used in the bygone years.

#### 3.1 Reconnaissance

Reconnaissance means gathering information about the potential target. A target can be an individual or an organizational entity. Reconnaissance can further be broken down to target identification, selection and profiling. Reconnaissance in cyber space mainly includes crawling World Wide Web such as internet websites, conferences, blogs, social relationship, mailing lists and network tracing tools to get information about the target. Information gathered from reconnaissance is used in later stages of cyber kill chain to design and deliver the payload. Reconnaissance is divided into 2 types [9] (expected to be done in order) (Table 1):

1. **Passive Reconnaissance:** This step is carried out by gathering the information about target without letting him/her know about it.
2. **Active Reconnaissance:** This step involves much deeper profiling of the target which might trigger an alert to the target.

**Table 1.** Reconnaissance Techniques

	Reconnaissance Methodology	Type of Reconnaissance	Techniques Used [10,11]
1	Target Identification and Selection	passive	Domain Names, whois, records from APNIC, RIPE, ARIN
2	Target Profiling		
	(a) Target Social Profiling	Passive	Social Networks, Public Documents, Reports and Corporate Websites
	(b) Target System Profiling	Active	Pingsweeps, Fingerprinting, Port Scanning and services
3	Target Validation	Active	SPAM Messages, Phishing Mails and Social Engineering

Reconnaissance provides knowledge about potential targets which will enable the attacker to decide the type of weapon suitable for the target, the types of delivery methods possible (Table 3), malware installation difficulties and security mechanisms that need to be bypassed. We will now see how the information gathered by reconnaissance is used to develop sophisticated malware.

## 3.2 Weaponize

Weaponize phase of the cyber kill chain deals with designing a backdoor and a penetration plan, utilizing the information gathered from reconnaissance, to enable successful delivery of the backdoor. Technically it is binding software/application exploits with a remote access tool (RAT). Weaponizing involves design and development of the following two components:-

### 3.2.1 RAT (Remote Access Tool)

RAT is a piece of software which executes on target's system and give remote, hidden and undetected access to the attacker. RAT is usually called the payload of a cyber-weapon. The target system can be a computer, mobile [13] or any embedded device provided the RAT software is properly compiled for the architecture being targeted. The types of access provided by a typical RAT (Table 2) are system exploration, file upload or download, remote file execution, keystroke monitor, screen capture, webcam or system power on/off with limited or user level privileges. If by some mechanism the RAT gets administrator/root access then it could include network spreading [1], network data capture access or persistent installation of anti-detection module. A RAT again constitutes of two major parts:

**Client.** Client is the piece of code which is delivered to the target, executes and creates a network connection with the Command and Control infrastructure of the RAT. After establishing connection, the client receives the command from its controller. Client in turn executes the command and sends the results back. It is not always necessary to have all RAT functionalities in a single deliverable. There are many instances of RAT like Carberp [14], Ventir Trojan [15], Poison Ivy [42], etc. where RAT functions are delivered to the target in a modular manner by using a very basic stub. The compiled code can be deployed in the form of an executable binary or shellcode. Shellcode in simplistic terms is a position independent machine code generally compiled by assemblers. There also exists techniques [16] which use specially crafted C-language source codes to generate shellcode.

**Server.** Server is the other half of RAT which runs on the Command and Control infrastructure with a nice UI (User Interface) displaying the connection information from target's client part. Server typically has options of commands like gathering keystrokes, file browser, screen capture, etc. Based on the objective of a cyber campaign, the attacker gives an appropriate command or series of

commands to the target using server interface which is executed by target's client side code and the output is returned back to the server. Depending on the execution permission level being allotted to the client part, these commands can be executed to get full access of target's system.

While developing RATs major constraints are size, anti-virus detection, extendibility, and scalability and user friendly interface. A big chunk of the RAT programing task is dedicated to UI design of server since ease of use of a RAT is defined by the server UI.

It is not necessary to code client and server in the same language. These days the servers are using either scripting languages(PHP, Javascript) for a web based UI [14,41] or C++, Java, delphi [17,42] for an application based interface.

**Table 2.** RATs and exploit kits

Famous RATs	Famous Exploit Kits [13,25]
Blackshades	Blackhole
DarkComet	Nuclear
Poison ivy	Styx
Bozok	Redkit
Njrat	Sweet Orange
Apocalypse	Infinity

### 3.2.2 Exploit

Exploit is the part of a weapon which facilitates the RAT to execute. Exploit acts as a carrier for the RAT and uses system/software vulnerabilities to drop and execute the RAT. The major objective in using exploits is to evade user attention while establishing a silent backdoor access using the RAT. Exploits can be of many forms like MS Office documents (.doc/ppt) [CVE-2010-3333, CVE-2014-4114] , PDF Documents [CVE-2014-9165, CVE-2013-2729], audio/video file [CVE 2013-3245] or web page [CVE-2012-1876, CVE-2014-6332] [18]. If a target opens any such file using the vulnerable software, RAT will be installed on targets system. After getting a RAT installed more exploits like privilege escalation exploits [CVE-2015-002, CVE-2013-3660] are used on the target to get higher privileges which can be used for spreading the RAT, persistent access and/or for destruction of the complete system.

There are methods to compromise the target using only RAT without using any exploit. But those methods are highly unreliable and not effective these days because of the increase in security awareness among the users. Embedding RAT in legitimate software executable, sharing via social engineering and faking RAT as genuine image/audio/video files are some of such methods which have been used in the wild in previous years [41].

### 3.3 Delivery

Delivery is the critical part of the cyber kill chain which is responsible for an efficient and effective cyber-attack. For any cyber-attack it is preferable to have target information to ensure a successful attack. In most of the cyber-attacks it is mandatory to have some kind of user interaction like downloading and executing malicious files or visiting malicious web pages on internet. This affinity comes from the information gathered during active and passive reconnaissance. There are some attacks which are performed without user interaction by exploiting network devices or services [CVE-2014-3306, CVE-2014-9583] [18].

Delivery is a high risk task for an attacker because delivery leaves traces. Therefore most of the attacks are performed anonymously using paid anonymous services, compromised websites and compromised email accounts.

While delivering the weapon multiple delivery methods are also used because no single method can guarantee 100 % success. Failed attacks sometimes are very useful to get basic information about the target's system information. Such types of information gathering mechanisms are very common in a browser based attack strategy, wherein a user visits the malicious web page which first tries to get user system information and then accordingly deliver the weapon [20].

**Table 3.** Delivery mechanisms

	Delivery Mechanism [1,12]	Characteristics
1	Email Attachments	Email content is composed to entice the user by using appealing content
3	Phishing Attacks	Sensitive information like usernames, passwords, credit card details etc. are extracted by masquerading a trustworthy entity in communication [19].
4	Drive by Download	Target is forced to download appealing malicious content from internet. Malicious content could be a image file, pdf/word document or software setup file
5	USB/Removal Media	Infected files are kept in Removable media which afterwards silently infects other systems opening the files.
6	DNS Cache Poisoning	Vulnerabilities in DNS are exploited to divert internet traffic from legitimate servers to attacker controlled destinations

### 3.4 Exploitation

After delivering the cyber weapon, then as expected the target completes the required user interaction and weapon executes at the target side. On execution, the next step is triggering the exploit. The objective of an exploit is to silently install/execute the payload. To trigger the exploit there are certain conditions that need to be matched:

1. User must be using the software/Operating System for which exploit has been created.
2. The software/Operating System should not be updated or upgraded to the versions wherein an exploit does not work.
3. Anti-Viruses or any other security mechanism should not detect the exploit or payload neither in statically nor dynamically scan during run time.

If all these conditions are fulfilled then exploit is triggered and will successfully install/execute the payload in target's system. Payload will connect to its Command and Control counterpart to inform about successful execution and wait for further commands to execute.

It is clear that exploit is the most critical part of the chain technically. What is an exploit and how does an attacker find such exploits is the next question here.

Exploits are made using vulnerabilities in software publicly identified as CVE [18]. Vulnerability is the software bug which can result in a potential threat to the system. A software bug is an unexpected condition in which a program/software misbehaves. Usually, while programming an application, lots of effort is made to avoid such conditions by writing every possible use case, handling exceptions for all types of inputs, validating inputs, etc. But most of the times not all cases are covered because of vast variety of user input/interaction. Therefore such invalid or wrong input forces the program to misbehave and this misbehavior is defined as type of vulnerability as mentioned in Table 4.

Vulnerabilities are further analyzed by exploit writers to see the possibility to carry and execute a payload. Not all vulnerabilities are exploitable. Some of them are just crashes or DoS(Denial of Service) or limited execution of program. It means not all vulnerabilities result into exploitable crashes.

Regarding the interest that how these vulnerabilities are discovered, there are papers [22–24] that describe details about fuzzing. Fuzzing is a methodology to give customized input to the program and monitoring the output for abnormal behavior. This abnormal behavior is further analyzed with respect to given input to describe vulnerability and so to create exploits.

As mentioned in the Delivery section sometimes just one delivery method is not sufficient, in the same way not just one exploit is sufficient to attack multiple users specifically during mass attacks. Generally combination of exploits, called an Exploit Kit, is used for this purpose. As the name suggests Exploit Kit is a collection of multiple exploits for various versions of software. As an example for browser based attacks an exploit kit (Table 2) may have exploits for various versions of Google Chrome, Firefox and internet explorer. During delivery if target uses vulnerable version of any of listed browser for which exploit is available in exploit kit it will be delivered accordingly.

### 3.5 Installation

Host based security measures have grown leaps and bounds compared to other security mechanisms. This in turn has spurred innovation in procedures that

circumvent host-based security controls to install, update and regulate the control of the malware installed upon the victims computing device. Traditionally, a computer would become infected by an infection vector like infected removable media, which in turn will leave a malware executable in some unusual location and modify registry/startup settings so that the malware executable is run every-time the computer boots up. Some user eventually will report [31] this executable to antivirus vendor, who in turn will analyze it and come up with a signature to detect it and in some cases a removal tool. Modern malwares are not that simple anymore [47]. Malware nowadays are multi staged and they heavily rely on droppers and downloaders to deliver the malware modules in a much more sophisticated manner.

- **Dropper** is a program that will install and run the malware to a target system. Before executing the malware code, dropper nowadays tries disabling host based security controls at the target and hides the installed malware.
- **Downloaders** were designed to perform the same actions as Droppers disabling the victims security and monitoring software, hiding core components and obfuscating the infection vector, etc. but tended to be smaller than Drop-pers because they did not contain the core malicious library components. Instead of unpacking an embedded copy of the core malware agent, the down-loader would connect to a remote file repository and download the core components.

Today’s installation life-cycle incorporates many checks, balances and resilience features as a means of maximizing the success of the installation, and protecting the participating attackers. Following are some techniques that mal-ware authors use for covert persistent and anonymous installations.

**Table 4.** Vulnerabilities and Exploits

	Category of Exploits	Type of Exploit	Type of Vulnerability [21]
1	Operating System Level Exploits	Kernel Exploits, Device Driver Exploits	<ul style="list-style-type: none"> <li>– Denial-of-Services</li> <li>– Remote or Local Code Execution</li> </ul>
2	Network Level Exploits	Protocol exploits for FTP, SMTP, NTP, SSH, Router exploits,	<ul style="list-style-type: none"> <li>– Privilege Escalation</li> </ul>
3	Application/ Software Exploits	Browser Exploits, MS Office exploits, PDF Exploits, Java/Flash Exploits	Memory Corruption: <ul style="list-style-type: none"> <li>– Dangling Pointer</li> <li>– Buffer Overflow [26]</li> <li>– Use-After-Free</li> </ul>



- **Anti-Debugger and Anti-Emulation:** Dropper and downloader components are typically armored. Using a variety of packers, crypters and inspection-detection engines, malware authors can ensure that common debugger and emulation analysis techniques will not work. The addition of advanced anti-virtual machine analysis technologies also deters malware analysis to a high extent.
- **Anti-Antivirus:** Many malware packages include toolsets for automatically disabling host-based detection technologies disabling anti-virus and IDS products installed on the victim's computer, changing local DNS settings to ensure that no future updates to the operating system or packages are possible, and adding tools that recheck and re-disable protection settings frequently.
- **Rootkit and Bootkit Installation:** Rootkits are programs which hide the executed payload. Payload file hiding, process hiding are the core functionalities of rootkits like LRK, AFX and Mebroot [27]. Similarly Bootkits are malwares that are able to hook and patch system to get loaded into the system Kernel, and thus getting unrestricted access to the entire system. Bootkits like Stoned Bootkit [28] modify the MBR or boot sector for its execution to avoid protections from operating system.
- **Targeted Delivery:** By performing a quick inventory of the victim's machine at the dropper/downloader stage and submitting this information to the malware distribution site, the attackers can verify that the compromised computer is real (and not some analysis system) and respond accordingly. In some cases, upon discovering that the victim was faked or is an automated analysis system, the attackers would not serve the core malware.
- **Host-Based Encrypted Data Exfiltration:** Most malware does not encrypt outbound network communication. Critical data stolen from the victim's computer is typically packed and file-encrypted at the host-level before sending over a clear text network protocol such as HTTP and SMTP thereby evading anomaly detection systems and data-leakage prevention systems (DLP) [47].

### 3.6 Command and Control

An important part of the remotely executed cyber-attacks is the Command and Control(C&C) system. C&C system is used to give remote covert instructions to compromised machines. It also acts as the place where all data can be exfiltrated. Over the years, the architecture of C&C channels have evolved exponentially owing to the exponential development of defensive mechanisms, namely antiviruses, firewalls, IDSs, etc. [29,32]. There are mainly three type of C&C communication structures, namely the traditional centralized structure, the newer peer-to-peer decentralized architecture and the latest Social Networks based structure.

- **Centralized Structure:** Traditionally malware depended upon a classical client server model wherein a central server is used to command and control the infected machines. Since there is only one server, its easy to manage. Also

there is no dependence on the infected machines to relay command control signals. Hence failure of random infected machines won't affect C&C architecture. But the number of bots that can be controlled depends is constrained by the hardware/software resources available to the C&C server. Also taking down the server strategically will shutdown the entire C&C infrastructure.

- **Decentralized Structure:** Owing to the fact that centralized architecture can be taken down easily and cannot control a large botnet [30], malware authors started using peer-to-peer P2P architecture for command and control. The main aims of using this architecture are scalability (infected machines are used as nodes, and each node in turn is responsible only for a subset of the total botnet), fault tolerance (redundant communication links can be formed to route information) and P2P nature (decentralized architecture removes the significant single point dependence of centralized architectures). Variety of options for P2P such as Bit torrent, Gnutella, LimeWire indicates the substantive depth of design possibilities. Eg. Storm [33]
- **Social Networks Based Structure:** Social networks now play a huge part in many peoples lives. Facebook has 1.35 billion registered users as of third quarter of 2014 [34]. Most of the social network services are free to use and these services are deemed to be benign in most organizational security policies. Owing to these reasons, social media has now become a viable option for malware authors. These high availability and reliable social networks are used to pass on information in a centralized/decentralized way to the infected machines. An example of such attacks is Taidoor [35].

C&C Communication Traffic analysis is a traditional technique to detect communication pattern among infected machines. As a result malware creators have adopted techniques to hide communication patterns. Anonymous communication techniques involve creating an unobservable communication channel that is resistant to traffic analysis. Unobservable communication channel refers to the communication capability that a third party cannot distinguish between a communication and non-communicating entity. It will be indistinguishable from legitimate traffic. Following is a review of techniques employed by the malware creators to achieve unobservable anonymous communication channel:

- **IRC Chats:** Internet Relay Chat(IRC) [36] was developed in 1988 and is a protocol used for text chat over the internet. Its primary function is to provide channels which are chat rooms allowing for public/private conversations. These channels were exploited by malwares to send and receive information once infection takes place.
- **TCP/HTTP/FTP:** Malwares relying on IRC as C&C channel were now easier to detect at network gateways. Malware authors circumvented this by employing TCP/HTTP/FTP [37–39] as C&C channels. Since majority of applications in all the operating systems use these channels for benign communication, malware traffic becomes indistinguishable. Examples: Zeus, Poison Ivy [41, 42]
- **Steganography:** Steganography technique involves hiding data inside images, video, audio or any such content which are hosted in attacker controlled harm-

less websites. The data is then hidden in specially constructed and encrypted inline annotations which are strewn randomly all over the page or extra spurious bytes of images. This hidden data can contain C&C commands like IP addresses of C&C server, shellcode commands to execute etc.

- **TOR:** Tor (originally The Onion Router) is a service used to provide anonymity over the Internet [40]. Tor directs Internet traffic through a free, worldwide, volunteer network consisting of more than six thousand relays to conceal a user's location and usage from anyone conducting network surveillance or traffic analysis. Apart from providing an anonymity network, Tor also has Hidden Service Protocol which makes it possible for users to hide their location while offering various kinds of public services. Hidden services work by setting up rendezvous points which are identified by .onion links. SDC Botnet [43] is a recent example of malwares employing TOR as C&C channel.

Attackers would like to stay undetected and covert for as long as possible. Following are 3 techniques commonly used by malware authors to hide their C&C server from the defense mechanisms of any organization:

- **DNS Fast Flux:** The Domain Name Service(DNS) [44] is a naming system of computers on the internet. DNS is not required to match hostnames to IP addresses in a one-to-one fashion. A single hostname may correspond to many IP addresses to facilitate fault tolerance and load distribution. Malware authors use a technique called fast-flux to hide C&C locations behind an rapidly changing(fast-flux) network of infected machines. This is achieved by setting a very short TTL(time-to-live) to DNS response which is known as single flux. Another technique called double flux is used which works by switching rapidly through multiple infected nodes addresses which in turn act as a proxy relay location to the original C&C site.
- **DNS as a Medium:** It is also possible to use DNS system as a communication mechanism rather than using it for setting up the channel. RDATA field of a DNS response can be of variable length and multiple formats. One of these is TXT, which can be used to transfer actual text. The commands are can be encoded into base64 and sent over the RDATA field. Feederbot [46] is one such example.
- **Domain Generation Algorithms:** Another method to stay under the radar is DGA. DGA programmatically generates pseudo random domain names which is used as address of C&C server. It is then upto the attacker to ensure he controls the domains that will be generated. Conficker malware for example, generates 250 domain names by using current UTC date as the seed of pseudo random DGA. The same domains are generated every 3 hours. All the 250 domain names are queried until one of them resolves successfully [45].

### 3.7 Act on Objectives

After getting the communication setup with target system, attacker executes the commands. The command used by attacker depends on interest of attack [48].

1. **Mass Attack:** Objective of mass attack is to get as many targets as possible. In mass attack more than a single system, multiple systems together are of interest. Most of such attacks aims at getting bank, email, social media and local system administrator credentials [49]. Bigger picture of mass attack is called BOTNets [50]. BOTNets are mainly used for DDoS attacks and virtual coin mining. Virtual coin mining harvest the system processors or GPU to generate virtual currency for the attacker.
2. **Targeted Attacks:** Targeted attacks are more sophisticated and carried with more caution. Most of such attacks are aimed to get confidential or secret information from target system. Data exfiltration and getting user credentials for online accounts are objectives of the attacks. Spreading through the network also becomes the primary goal when target is an organization.

In both types of attack if attack is intended for destructive purpose it may crash system hard drive or device drivers. Attacker may make CPU to use its maximum capability for long time to damage the processor hardware.

## 4 Summary

In this paper cyber kill has been discussed in detail with its technical aspects. The paper gives technical flow from attacker's perspective which may help security researcher to design prevention mechanisms. This paper explores the recent trends in malware development by reviewing weaponize and installation techniques. Delivery and Exploitation section gives an insight to the importance of software vulnerabilities, both from an attacker and security researcher's perspective, in the current cyberspace. Study of reconnaissance techniques and Command and Control infrastructure shows how benign features of network protocols are being misused to achieve nefarious end results.

## References

1. Malware Risks And Mitigation Report. 1st ed. BITS - The financial services roundtable (2011). <http://www.nist.gov/itl/upload/BITS-Malware-Report-Jun2011.pdf>
2. Ranum, M.J.: Breaking Cyber Kill Chains. Tenable Network Security (2014). <http://www.tenable.com/blog/breaking-cyber-kill-chains>
3. Sager, T.: Killing Advanced Threats in Their Tracks: An Intelligent Approach to Attack Prevention. Sansorg (2014). <http://www.sans.org/reading-room/whitepapers/detection/killing-advanced-threats-tracks-intelligent-approach-attack-prevention-35302>
4. The Cyber Attack Cycle. [http://www.eur.army.mil/vigilance/Cyber\\_Attack\\_Cycle.pdf](http://www.eur.army.mil/vigilance/Cyber_Attack_Cycle.pdf)
5. Hartley, M.: Strengthening Cyber Kill Chain with Cyber Threat Intelligence. iSIGHT Partners (2014). <http://www.isightpartners.com/2014/09/strengthening-cyber-kill-chain-cyber-threat-intelligence-part-1-of-2/>

6. Hartley, M.: The Cyber Threat Kill Chain Part 2 of 2 - iSIGHT partners. iSIGHT partners (2014). <http://www.isightpartners.com/2014/10/cyber-threat-kill-chain-part-2-2/>
7. Davis, R.: Exploit Kill Chain with Controls — Critical Start. Criticalstartcom (2015). <http://www.criticalstart.com/2014/01/exploit-kill-chain-with-controls/>
8. Engel, G: Deconstructing the Cyber Kill Chain. Dark Reading (2014). <http://www.darkreading.com/attacks-breaches/deconstructing-the-cyber-kill-chain/a/d-id/1317542>
9. IT Security Reconnaissance. <http://itsecurity.telelink.com/reconnaissance/>
10. Pernet, C.: APT Kill chain - Part 3: Reconnaissance - Airbus D&S CyberSecurity blog (2014). <http://blog.cassidiancybersecurity.com/post/2014/05/APT-Kill-chain-Part-3-Reconnaissance>
11. Bhamidipati, S.: The Art of Reconnaissance - Simple Techniques. sans.org (2002). <http://www.sans.org/reading-room/whitepapers/auditing/art-reconnaissance-simple-techniques-60>
12. Security Threat Report 2013. 1st ed. SOPHOS. <http://www.sophos.com/en-us/medialibrary/pdfs/other/sophossecuritythreatrep-ort2013.pdf>
13. Security Threat Report 2014. 1st ed. SOPHOS. <https://www.sophos.com/en-us/medialibrary/PDFs/other/sophos-security-threat-report-2014.pdf>
14. Rodionov, E., Matrosov, A.: Defeating Anti-Forensics in Contemporary Complex Threats
15. Securelist.com.: The Ventir Trojan: Assemble Your MacOS Spy - Securelist. N.p (2015)
16. Anley, C., et al.: The Shellcoder's Handbook: Discovering and Exploiting Security Holes. John Wiley & Sons, New York (2011)
17. Research.zscaler.com.: Zscaler Research: Njrat & H-Worm Variant Infections Continue To Rise. N.p (2015)
18. CVE -Common Vulnerabilities and Exposures (CVE). <https://cve.mitre.org/>
19. Pernet, C.: APT Kill chain - Part 4: Initial compromise - Airbus D&S CyberSecurity blog 2014. <http://blog.airbuscybersecurity.com/post/2014/06/APT-Kill-chain-Part-4-%3A-Initial-compromise>
20. GitHub, DeviceFingerprint. <https://github.com/dimalinux/DeviceFingerprint>
21. CVE security vulnerability database. Security vulnerabilities, exploits, references and more. <http://www.cvedetails.com/>
22. Oehlert, P.: Violating assumptions with fuzzing. IEEE Secur. Priv. **3**(2), 58–62 (2005)
23. Sutton, M., Greene, A., Amini, P.: Fuzzing: Brute Force Vulnerability Discovery. Pearson Education, Upper Saddle River (2007)
24. Godefroid, P., Levin, M.Y., Molnar, D.: SAGE: whitebox fuzzing for security testing. Queue **10**(1), 20 (2012)
25. Contagio: An Overview of Exploit Packs (Update 24), March 2015. <http://contagiodump.blogspot.in/2010/06/overview-of-exploit-packs-update.html>
26. Chien, E., Sze, P.: Symantec Security Response: Blended Attacks Exploits, Vulnerabilities and Buffer-Overflow Techniques in Computer Viruses Virus Bulletin (2002). <http://www.symantec.com/avcenter/reference/blended.attacks.pdf>
27. Hardikar, A.: Malware 101 - Viruses. sansorg (2008). <http://www.sans.org/reading-room/whitepapers/incident/malware-101-viruses-32848>
28. Kleissner, P.: Stoned bootkit. In: Black Hat, USA, pp. 5–7 (2009)
29. Gradiner, J., Cova, M.: Shishir Nagaraja: Command and Control : Understanding, Denying and Detecting (2014)

30. Seenivasan, D., Shanthi, K.: Categories of botnet: a survey. *Int. J. Comput. Control Quantum Inf. Eng.* **8**(9), 1589–1592 (2014)
31. Yen, T.-F., Heorhiadi, V., Oprea, A., Reiter, M.K., Juels, A.: An epidemiological study of malware encounters in a large enterprise. In: *ACM SIGSAC Conference on Computer and Communications Security* (2014)
32. QinetiQ. Command & Control: Understanding, Denying, Detecting, 36 February 2014. <http://www.cpni.gov.uk/Documents/Publications/2014/2014-04-11-cc/qinetiq/report.pdf>
33. Porras, P., Saidi, H., Yegneswaran, V.: A multi-perspective analysis of the Storm (Peacomm) worm. In: *SRI Technical Report 10-01* (2007)
34. Statista. Facebook: figures of monthly active users 2014 — Statistic (2015). <http://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/>
35. Fireeye.com. Evasive Tactics: Taidoor Threat Research — FireEye Inc. (2013). <https://www.fireeye.com/blog/threat-research/2013/09/evasive-tactics-taidoor-3.html>
36. Tools.ietf.org. RFC 1459 - Internet Relay Chat Protocol (2015). <https://tools.ietf.org/html/rfc1459>
37. Ietf.org. RFC 2616 - Hypertext Transfer Protocol - HTTP/1.1 (1999). <https://www.ietf.org/rfc/rfc2616.txt>
38. Ietf.org. RFC 959 - FILE TRANSFER PROTOCOL (FTP) (1985). <https://www.ietf.org/rfc/rfc959.txt>
39. Ietf.org. RFC 793 - TRANSMISSION CONTROL PROTOCOL (1981). <https://www.ietf.org/rfc/rfc793.txt>
40. Fox-IT International blog. Large botnet cause of recent Tor network overload (2013). <http://blog.fox-it.com/2013/09/05/large-botnet-cause-of-recent-tor-network-overload>. Accessed 24 March 2015
41. IOActive Inc. Reversal and Analysis of Zeus and SpyEye Banking Trojans (1st ed., p. 31). Seattle: IOActive, Incorporated (2012). <http://www.ioactive.com/pdfs/ZeusSpyEyeBankingTrojanAnalysis.pdf>
42. FireEye. Poison Ivy: Assessing Damage and Extracting Intelligence (1st ed., p. 33). California: FireEye Inc (2014). <https://www.fireeye.com/resources/pdfs/fireeye-poison-ivy-report.pdf>
43. ydklijnsma: Large botnet cause of recent Tor network overload 2013. <http://blog.fox-it.com/2013/09/05/large-botnet-cause-of-recent-tor-network-overload/>
44. Ietf.org. RFC 1035 - DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION (1987). <https://www.ietf.org/rfc/rfc1035.txt>
45. Porras, P., Saidi, H., Yegneswaran, V.: A foray into confickers logic and rendezvous points. In: *Proceedings of the USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)* (2009)
46. Dietrich, C.J., Rossow, C., Pohlmann, N.: CoCoSpot: clustering and recognizing botnet command and control channels using traffic analysis. *Comput. Netw.* **57**(2), 475–486 (2013)
47. DAMBALLA. Behind Today's Crimeware Installation Lifecycle: How Advanced Malware Morphs to Remain Stealthy and Persistent 10 (1st ed., p. 10). Atlanta: DAMBALLA (2015). [https://www.damballa.com/downloads/r-pubs/WP\\_Advanced\\_Malware\\_Install\\_LifeCycle.pdf](https://www.damballa.com/downloads/r-pubs/WP_Advanced_Malware_Install_LifeCycle.pdf)
48. A View From Front Lines. 1st ed. MANDIANT A FireEye Company (2015). <http://www2.fireeye.com/rs/fireeye/images/rpt-m-trends-2015.pdf>

49. Pernet, C.: APT Kill chain - Part 5: Access Strenghtening and lateral movements - Airbus D&S CyberSecurity blog (2014). <http://blog.airbuscybersecurity.com/post/2014/11/APT-Kill-chain-Part-5-3A-Access-Strenghtening-and-lateral-movements>
50. Naseem, F., shafqat, M., Sabir, U., Shahzad, A.: A survey of botnet technology and detectiion. Int. J. Video Image Process. Netw. Secur. IJVIPNS-IJENS **10**(01), 9–12 (2010)