

# Literature Review

Jordan Peters

March 1, 2020

I certify that all material in this dissertation which is not my own work has been identified.

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Literature Review</b>	<b>2</b>
2.1	What do researchers know? . . . . .	2
2.1.1	Software Vulnerabilities . . . . .	2
2.1.2	System Level Vulnerabilities . . . . .	3
2.2	What do they not know? . . . . .	3
2.3	What has been researched and what has not been researched? . . . . .	3
2.4	Is the research reliable and trustworthy? . . . . .	3
2.5	Where are the gaps in the knowledge? . . . . .	3
<b>3</b>	<b>Conclusion</b>	<b>3</b>
<b>4</b>	<b>Bibliography</b>	<b>3</b>

# 1 Introduction

In 2010, a "worm" virus, one that spreads across and embeds itself into systems, infected approximately 100,000 systems [1] and controlled 984 nuclear centrifuges to repeatedly malfunction them [2] over the span of at minimum 1 year, and likely more. [1] It eluded detection by performing specific subroutines that would cause the equipment to only breakdown in such a way that it would cause no harm to people, and would make the scientists believe the equipment they were sold was just faulty and that they were unlucky. [2]

This virus was called Stuxnet, and its inception 10 years ago caused many researchers to become scared at the real-world implications that Stuxnet showed. [1] [2] Stuxnet proved that cybernetic attacks on critical infrastructures such as nuclear reactors are possible, and aren't just the type of attack that exists within the realms of theory or movie plotlines. [1] Stuxnet was of such high complexity and danger that security researchers at Symantec said they hope to never see anything like this again. [1]

How do attacks like these ever get remotely close to the systems that they target? What are their routes of intrusion? Was it difficult for the attackers to attack this way, or is it easy if you have the knowhow? This literature review aims to answer the question of how cybersecurity penetration is often orchestrated, and what researchers are attempting to do to analyse these attack vectors and deny infection altogether before catastrophe can be inflicted.

## 2 Literature Review

### 2.1 What do researchers know?

Attack vectors, defined as the path used by an attacker to conduct an attack or to attempt to bypass system controls, [3] today are increasing as a result of the emerging nature of new technologies. As almost all of software developer's focus is put on developing software to solve a business problem, the security of this software becomes inherently neglected. The multi-contextual nature of software and its very large plethora of applications mean that the number of attack vectors attackers can choose to move through increase. These are commonly referred to in the same context as attack surfaces, known as the amount of ways an attacker can attack a given system. The greater the attack surface, the more vulnerable the system.

Stuxnet isn't fair to compare to the rest. It utilised 4 zero-day vulnerabilities, known as a vulnerability that was not previously known to the public, and used 2 stolen digital certificates, a security mechanism used to secure communications between two machines over the Internet, [1] to complete its task of disrupting operations at Iran's illicit nuclear research facility. [2] This is not normal. Hackers don't reveal more than one zero-day unless they have to, and stolen digital certificates are almost never used in an attack due to the sheer difficulty of stealing the certificates physically from the companies to be capable of it. [2] A willing or unwilling contractor or insider for the Iran facility was likely used to carry out the attack, and another insider was likely used to steal the digital certificates from the certificate authority companies. [1] This was a marksman's job from beginning to end and does not compare to ordinary cyber attacks today. [1]

#### 2.1.1 Software Vulnerabilities

This does provide us a glimpse into the common use of a **combination** of attack strategies to accomplish a goal. Here we see the use of zero-day vulnerabilities, through a type of attack vector known as software vulnerabilities, where bugged code in software is exploited. [2] For Stuxnet, vulnerabilities in antivirus software were exploited, along with system level vulnerabilities discussed in 2.1.2. During the virus' initial injection, it would scan the currently running antiviruses and system processes, read metadata like their version number, and decide which process to inject its malicious code into, optionally using one of two zero-day vulnerabilities for code injection that it contained. [1] The virus does this to elevate privileges so it can perform any action it desires on the target computer. [1] If no process was bypassable, the virus would become inert. [1]

### 2.1.2 System Level Vulnerabilities

## 2.2 What do they not know?

## 2.3 What has been researched and what has not been researched?

## 2.4 Is the research reliable and trustworthy?

## 2.5 Where are the gaps in the knowledge?

## 3 Conclusion

Summary and conclusions to take forward.

## 4 Bibliography

### References

- [1] N. Falliere, L. O. Murchu, and E. Chien, “W32. stuxnet dossier,” *White paper, Symantec Corp., Security Response*, vol. 5, no. 6, p. 29, 2011. [Online]. Available: <https://nsarchive2.gwu.edu/NSAEBB/NSAEBB424/docs/Cyber-044.pdf>
- [2] P. W. Singer and A. Friedman, *Cybersecurity: What everyone needs to know*. oup usa, 2014. [Online]. Available: [https://books.google.co.uk/books?id=B88ZAgAAQBAJ&printsec=frontcover&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](https://books.google.co.uk/books?id=B88ZAgAAQBAJ&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)
- [3] C. Roberts, “Biometric attack vectors and defences,” *Computers & Security*, vol. 26, no. 1, pp. 14–25, 2007.