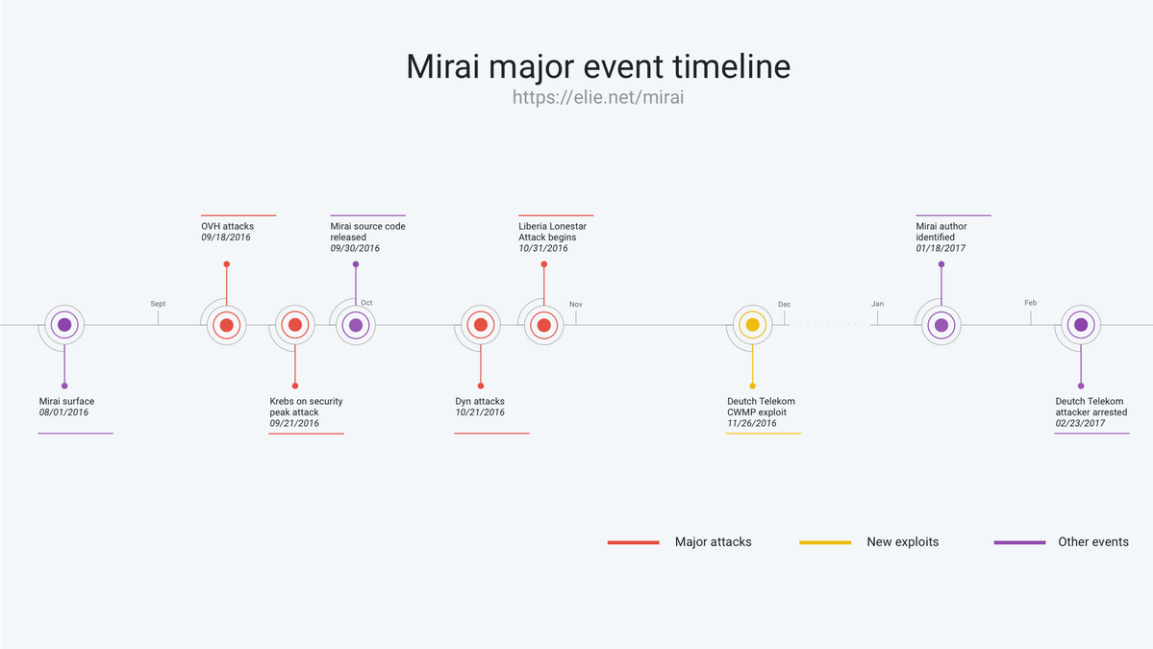
Northern Arizona University

Cybersecurity
CYB 410 - Software Security

Analyzing and Reporting on Stuxnet

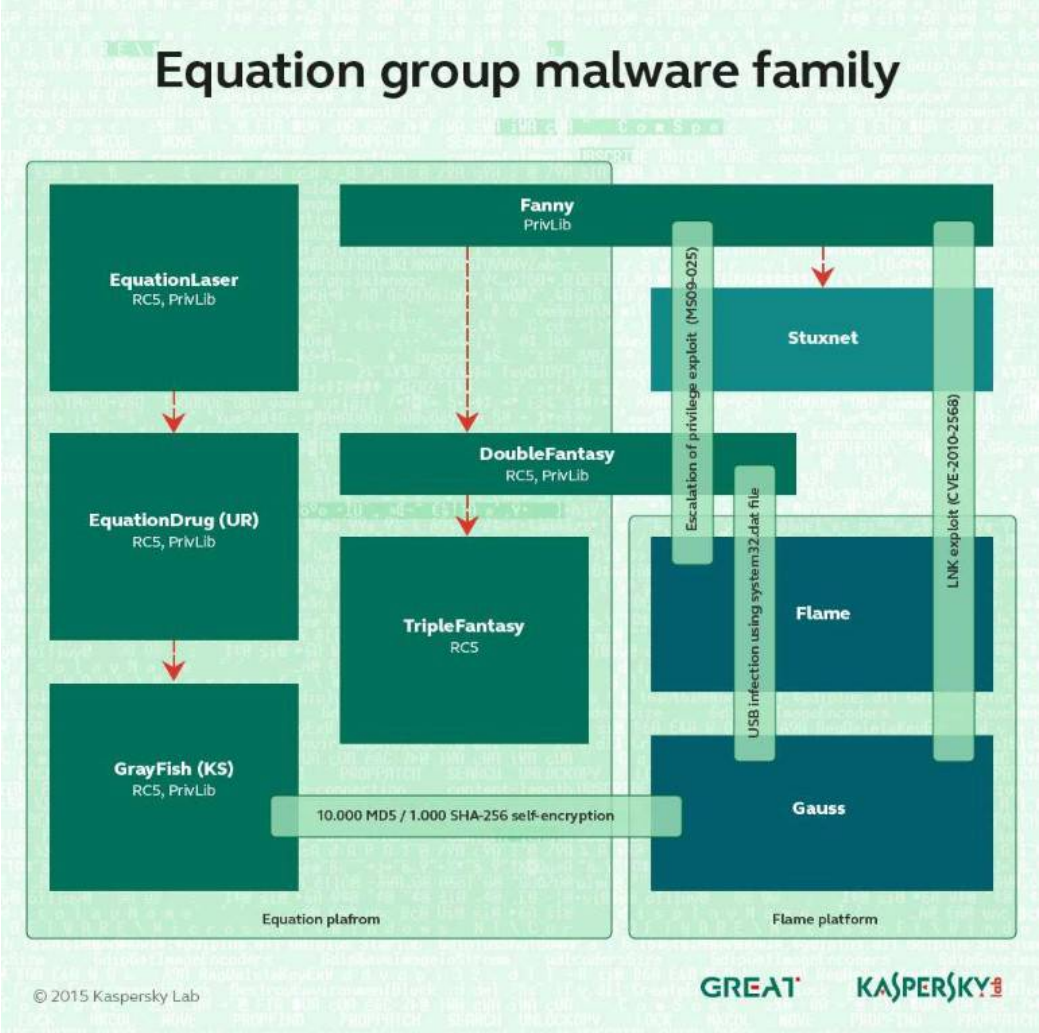


Author:
Akiel Aries

Professor:
Prof. Sareh Assiri

November 20, 2022

1 Overview



Stuxnet is a computer worm first identified in 2010 that was originally implemented to target Iranian governments nuclear facilities. Just like most virus', they mutate and spawn new strains often more dangerous than their predecessor. The original infection succeeded in targeting PLCs (Programmable Logic Controller) which are used in a plethora of manufacturing related processes. For example, PLCs are used in robotic machinery seen in assembly lines at manufacturing plants. The point of these controllers is to be secure, reliable, and useful fault diagnostics.
It is no wonder why Stuxnet was a large threat in the InfoSec industry. It is of popular belief that this is the first computer worm capable of affecting hardware and its peripherals as most computer worms are built as software exploits. The origin of the worm is cloudy however, according to reports from cybersecurity research firms such as Kapersky Lab, Symantec, and many more, the origin supposedly started as a nation-based collaboration with the United States and Isreal to target the Iranian Nuculear Program. The worm went on to destroy 1/5 of Iran's nuclear centrifuges, an important component in the process of enriching

collected uranium, infecting above 200,000 thousand devices and went on to severely degrade the performance of over 1,000 machines themselves. What lead researchers to believe the bug was nation state sponsored was the lack of infections of other users of the Seimens based software. This was false according toe Eugene Kapersky himself as he said a Russian neuclear power plant was also infected but not affected since it lacked any access to the public-facing internet.
The repository that this report will be base the code off of is located here: https://github.com/research-virus/stuxnet

2 Source Code

Scale

Execution

Reproducible Example

Static Analysis/Debugging

Fuzzing with AFL

Final Notes