Authorship analysis of the Zeus botnet source code

—

Presented by -
Payal Singh

Zeus, ZeuS or ZBot

A malware that installs onto a person's computer, joining it to a large botnet capable of stealing information, sending spam and phishing, and performing attacks to infect other computers. Zeus employs stealth techniques, such as obfuscation and packing, to make it difficult to detect even if the signature is known.

In 2011 the source code to Zeus version 2.0.8.9 was publicly released.

What insights do authorship and profiling lead to?

How big are the groups that perform cybercrimes?

Who should respond to a particular cyberattack?

Was it a criminal act, an act of war, or just a cyber- based vandalism?

Can we expect further attacks, or was this a one-off?

Authorship as a response to cybercrimes

Increased risk to offenders, not just detecting and limiting profits

Increased defenses when moving from commodity to specialized malware

Contributions

Systematic and automated method for Zeus source code analysis

Number of authors

Zeus source code characteristics

Stealth

- Encryption
- Obfuscation

Modularization

- Keyloggers
- Web injection
- Snapshots

Methodology

Evidence Accumulation Clustering (EAC)

- Unsupervised initial clustering
- Automated final clustering

Manual Analysis

Initial Clustering

Recentred Local Profiles (RLP)

- Takes the frequency of top L most frequently occurring n-grams, using distinctiveness as a measure
- Distinctiveness - recentering frequencies of n-grams

Document comparison

- cosine distance over union of features common to both documents

Inverse-author-frequency in RLP

Final Clustering

Coassociation matrix C with hierarchical clustering

Dendogram as output

Primarily braces and comments

Noise

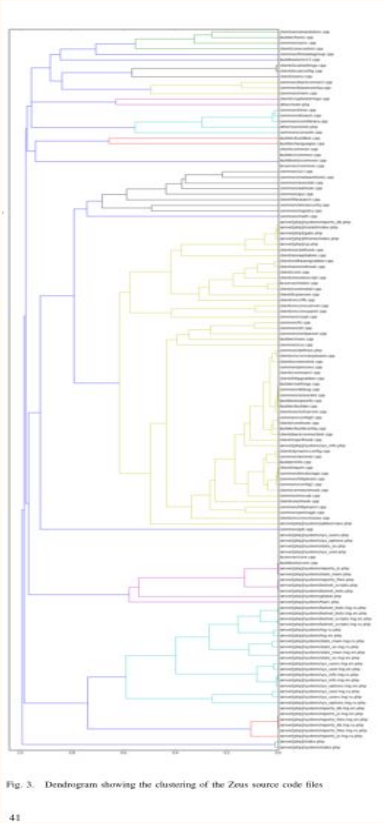

Fig. 3. Dendrogram showing the clustering of the Zeus source code files

41

Findings

Primarily two authors, additional authors on smaller sections

Each author wrote whole functions, but each file has functions written by multiple authors

Limitations

Confirmation - cannot be confirmed

Applicability - Cannot be applied to determine the attacker, only the creator

Multi-authored files - Different authors work on same files across versions

Noise - lack of comments and structure lead to lack of identifiable characteristics

Relative attribution - difference and similarities, not a definite name

Scaling - requires manual intervention

Future work

Splitting single files into multiple parts for intra-file analysis

Searching for similarities with open source projects

Critique

Requires proofreading - typos and spelling mistakes

No new technique - using already available algorithms, no change

Lack of details - could have elaborated more on which features they looked for

Thanks!

Questions?