Demo of Slips, a Free-Software IPS with Behavioral Machine Learning Detection

Sebastian Garcia, Kamila Babayeva, Alya Gomaa, Ondrej Lukas

***Abstract*—Slips is a behavioral-based Python IDS, and the first free software IDS that uses machine learning to detect malicious behaviours in the network traffic. It creates behavioral profiles of IP addresses and performs detection on those profiles based on time windows. Its modular architecture allows for the combination of traditional approaches, such as Threat Intelligence, and rule-based detections, together with machine learning methods. Slips is able to detect various types of attacks and visualize the detections using GUI Kalipso. In this demo, we demonstrate the usage of Slips on the network traffic of a real Android Remote Access Trojan attack, and show how it can be used for analysing the traffic of such advanced threats.**

***Index Terms*—Intrusion Detection Systems, Security and Privacy, Machine Learning**

I. Introduction

Applying machine learning methods for detections in the network traffic is a challenging task. The research on machine learning for network security has been mostly implemented inside companies, effectively stopping millions of people from using this type of technology to protect themselves for free.

Our solution for this problem was the development of Slips, a behavioral-based Python IDS that uses machine learning to detect malicious behaviour in the network traffic. Slips is a free software tool that can detect attacks, scans, malicious attackers and command and control channels from malware, providing at the same time a good visualization for the analyst. The free software code of Slips can be found in https://github.com/stratosphereips/StratosphereLinuxIPS.

II. Slips Architecture

Slips was designed to focus on targeted attacks, command and control channels and to provide good visualizations. This includes being able to be updated and extended easily.

A. Core

The core of Slips consist of a large range of input sources, which are all interpreted to obtain a normalized version of the features. This input is processed into profiles for each IP address, with all the data being stored in a Redis database. This Redis database acts as a central repository of all Slips knowledge and decisions. The modularity is implemented by loading python modules developed by users. All these modules interact with the Redis database, augmenting the decisions, detections and outputs. The modularity of the software allows for easy extension and fast prototyping. When Slips is executed, it spawns several child processes to manage the I/O, to profile attackers and to run the detection algorithms. Finally, Slips has many output types and one user interface.

S. G. is in the Stratosphere Laboratory, in the Czech Technical University in Prague, sebastian.garcia@agents.fel.cvut.cz

K. B. is in the Stratosphere Laboratory, and EPFL, kamifai14@gmail.com

A. G. is in the Stratosphere Laboratory, and in Mansoura University, alyaggomaa@gmail.com

O. L. is in the Stratosphere Laboratory, in the Czech Technical University in Prague, lukasond@fel.cvut.cz

B. Input

Slips is designed to work with various inputs types. It can read the packets directly from an interface, and it can digest outputs from other tools such as Suricata.

All input data is converted into a unique internal format. Data formats supported by Slips are listed below.

- Pcap files (internally using Zeek [3])
- Packets directly from an interface (internally using Zeek)
- Suricata [4] flows (from JSON files created by Suricata, such as eve.json)
- Argus flows (CSV file separated by commas or TABs)
- A Zeek folder with flow files and a unique conn.log file alone.
- Nfdump flows from a binary nfdump file

C. Internal representation of data

Slips works at a flow level, instead of a packet level, gaining a high level view of behaviors. A profile is created for each IP that appears in the traffic. Each profile contains the complete behavior of the IP address. Profiles are further divided into time windows (by default 1 hour long). Detection modules can classify each of the profile in each time window and store the results together with the supporting evidence in the database.

D. Usage of Zeek

Slips uses Zeek for reading packets from the interfaces and pcap files, and as additional source of data for creating the profiles. All Zeek log files, such as DNS and HTTP, are used

for enriching the behavioral profiles, and creating a visual timeline of the profile activity in each time window. All files are read in real time, and sorted automatically so the timeline has all the flow types in order. E.g. the TLS flows after the DNS flow that resolved the IP address.

E. Usage of Redis Database

Slips stores data in two Redis databases, so modules access data in parallel. Slips also uses the Redis messaging system called Redis PUB/SUB. Modules publish data into the channels, while others subscribe to these channels and receive data. One database is used for the current data, and the other is used for a cache.

F. Slips Detection Modules

Each module is independent, waiting data from the channels and stores back in Redis. The current modules are:

- ASN: Get the Autonomous System Number of each IP
- geoip: Finds the country of each IP
- port scan: Detects Horizontal and Vertical port scans
- threat Intelligence - Checks 50 TI feeds for IP, and Domains
- timeline: Creates a timeline of flows
- blocking: Blocks malicious IP addresses
- flowalerts: Detects attacks in flows (self-signed certificate, long duration connections, etc.)
- rnn-cc-detection: Detects C&C channels using recurrent neural networks
- RDNS: Gets the reverse DNS of each IP
- http-analyzer: Detects attacks in HTTP
- VirusTotal: Get IP, domains, URLs on VirusTotal [5]
- Whitelist: Ignores organizations, IPs or domains

G. Stratosphere Behavioral Letters

Slips gets all the flows going to the same destination IP, port and protocol, and uses a symbol per-flows to show their characteristics. Each flow is assigned a letter based on its duration, size, and periodicity, creating a string. This string characterizes the behavior of the connection and allows a better detection. Figure 1 shows the matrix used for the letters based on size, duration and periodicity of each flow.

H. Kalipso

Kalipso is a command-line, Node.js, GUI designed for Slips. It gives users an overview of the data, attacks and malicious behaviors that were detected. Kalipso interface is shown in Figure 2.

	Size Small			Size Medium			Size Large		
	Dur. Short	Dur. Med.	Dur. Long	Dur. Short	Dur. Med.	Dur. Long	Dur. Short	Dur. Med.	Dur. Long
Strong Periodicity	a	b	c	d	e	f	g	h	i
Weak Periodicity	A	B	C	D	E	F	G	H	I
Weak Non-Periodicity	r	s	t	u	v	w	x	y	z
Strong Non-Periodicity	R	S	T	U	V	W	X	Y	Z
No Data	1	2	3	4	5	6	7	8	9

Symbols for time difference:

Between 0 and 5 seconds:	.
Between 5 and 60 seconds:	,
Between 60 secs and 5 mins:	+
Between 5 mins and 1 hour:	*
Timeout of 1 hour	0

Fig. 1. Slips behavioral letters describing the behaviour of flows in size, duration and periodicity.

III. Detection of a RAT using Slips and Kalipso

To show the detection capabilities of Slips and Kalipso, we analyze a real malware infection by the DroidJack v4.4 [6] for Android. DroidJack is a functional remote access trojan (RAT) used to spy on mobile phones.

The capture contains both normal and malicious traffic. During the RAT execution the Android APK was configured with the controller IP **147.32.219.67**, port **1337** as C&C, and port **1334** as a default port in the RAT configuration. First, run Slips and Kalipso on the pcap file:

```
./slips.py -f <pcap_name> -G
```

where -f is the pcap input file and -G starts the Kalipso interface.

Kalipso starts by showing the main window as in Figure 2. It displays all the profiles created, the time windows, the timelines for each time window, information for a selected IP and evidence found in each time window. Infected time windows are highlighted in red.

The left column in Figure 2 shows the client IP 10.8.0.37 and its first time window In the right column Kalipso shows all the information regarding all the activity in the mobile device as a timeline and detections performed by Slips.

The box with the evidence at the bottom of Figure 2 displays the detections performed by Slips (Figure 3). Since the first part of the traffic is normal, there is a series of detections from the threat intelligence feeds - connections to advertisement servers and trackers.

```
Detected dstdomain www.googleadservices.com. Blacklisted in {"description": "2021/5/5", "source": "adserversandtrackers.csv"}
Detected dstdomain app-measurement.com. Blacklisted in {"description": "2021/5/5", "source": "adserversandtrackers.csv"}
Detected dstip 147.32.83.253. Multiple reconnection attempts to Destination IP: 147.32.83.253 from IP: 10.8.0.57
Detected dstip 147.32.83.253. Connection to multiple ports ['1334', '1337'] of Destination IP: 147.32.83.253
```

Fig. 3. Kalipso evidence box when analyzing DroidJack v4.4 network traffic.

Next two pieces of evidence in Figure 3 are reconnection attempts to the C&C 147.32.83.253. It is not common for the devices to connect several times to the closed destination port, but RATs tend to do this forever. Moreover, the infected device connects to the ports 1334 and 1337 of the C&C. Both ports are running unknown services, and were detected by Slips.

Kalipso has a number of widgets that gather information about the connections. In Figure 4 Kalipso shows the table

Fig. 2. Main page of the Kalipso GUI after analysing the DroidJack v4.4 RAT capture.

with detailed information about each connection from the device, including VirusTotal scores, country, ASN and resolved DNS domain. In particular, the second column from the left shows the behavioral strings of all connections from the device IP 10.8.0.37 during time window 1.

Fig. 4. Kalipso shows information about the connections from the device to different IPs and ports (aggregated flows), including the behavioral strings (2nd column), DNS (3rd column), SNI (4th column), ASN (5th column), country (6th column), and VirusTotal scores (7th - 10th columns). All this information is only for the current time window analyzed.

There are several connection to the C&CL: 147.32.83.253 port 1337/TCP, 147.32.83.253 port 1334/TCP, and 147.32.83.253 port 1337/UDP.

The behavioral model of the connection to 147.32.83.253 on port 1337/UDP represents the behaviour of the C&C. According to the behavioral letters shown in Figure 1, the letter 'a' stands for a flow with strong periodicity, short duration and small size.

We obtained the following information:

1) Connections to uncommon ports 1337 and 1334.
2) The connection to 147.32.83.253 port 1337/UDP is periodic and defines the connection to the C&C server.
3) The phone continually reconnects to the IP 147.32.83.253, which is not a common behaviour. This behaviour was detected by the tool.

IV. Conclusion

This demo introduced a free software behavioral-based Python IDS called Slips, and showed how it can analyze and detect a real Android malware network infection. Additionally, we have presented Kalipso GUI for Slips, a tool for assisting the analysts and visualizing the detections.

Using machine learning and other techniques, Slips adapts to new malware attacks. It is possible for a non-expert to understand the detections and act. As for the analysts, Slips facilitates the analysis enormously by pointing that the device is infected and it requires a deeper analysis. The code of Slips is free-software and can be downloaded from https://github.com/stratosphereips/StratosphereLinuxIPS.

References

[1] Stratosphere, Stratosphere Laboratory Datasets, 2015, https://www.stratosphereips.org/datasets-overview. Accessed 11.8.2021
[2] Redis,Redis Labs, https://redis.io/, Accessed 10.8.2021
[3] Zeek, International Computer Science Institute (ICSI), https://zeek.org/. Accessed 11.8.2021
[4] Suricata, Open Information Security Foundation (OISF), https://suricata.io/. Accessed 11.8.2021
[5] Virustotal,Virustotal Labs, https://virustotal.com/. Accessed 11.8.2021
[6] Android Mischief Dataset v2: DroidJack v4.4., Kamila Babayeva Stratosphere Laboratory, https://mcfp.felk.cvut.cz/publicDatasets/Android-Mischief-Dataset/AndroidMischiefDataset_v2/RAT02_DroidJack/. Accessed 11.8.2021