Zeek

Table of contents:

Part 1. About zeek.

Zeek is an open-source network traffic analyzer that was created in 1994 by Vern Paxson and was called at first "Bro".

Zeek is not a standard intrusion detection system (IDS), Zeek can support investigation of suspicious or malicious traffic, also Zeek support performance measurement and troubleshooting analysis.

Zeek is easy to use traffic analyzer because its generating log files with reach information on each packet logged in the pcap file therefore generating better understanding of network traffic and usage.

Part 2. Comparison to similar software.

	Zeek	Snort	Suricata	Ossec
Signature-based IDS	V	V	V	V
Anomaly-based IDS	V	X	X	X
Scriptable configuration	V	V	V	X
Easy to set up	X	V	V	V
File Extraction	V	X	V	X
Multiple protocols support	V	V	V	V
Cross platform	X	V	V	V
Community support	V	V	V	V
File integrity monitoring	X	X	X	V
Automation capabilities	V	X	X	V

Part 3. Zeek pros and cons.

Pros:

- Zeek have wide range of traffic analysis tasks beyond the security domain.

- Zeek shows big range of logs describing network activity and can show application-layer transcripts as well.

- Zeek writes all the information into well-structured tab-separated or JSON log files suitable for post-processing with external software.

- Zeek is customizable and extensible platform for traffic analysis.

- Zeek runs on commodity hardware.

- Zeek scripting language facilitates a much broader spectrum of very different approaches to finding malicious activity semantic misuse detection, anomaly detection, and behavioral analysis.

- Zeek cluster features support single-system and multi-system setups.

Cons:

- Zeek is not optimized for writing traffic to disk, and need another program to do so.

- Zeek is not a protocol analyzer and not seeking to depict every element of network traffic at the frame level.

- Zeek don't have high level signature detection. (such as Suricata/ Ossec)

- Zeek have low level capability to preform byte-centric intrusion detection.

Part 4. Zeek installation and preparation.

The following installation for zeek will be done on ubuntu server x.

1. sudo su (-to save time)

2. check that you are in the desktop folder (- so every step will be the same as I did)

3. apt-get install cmake make gcc g++ flex bison libpcap-dev libssl-dev python3 python3-dev swig zlib1g-dev

4. apt-get install python3-git python3-semantic-version

5. git clone --recursive https://github.com/zeek/zeek

6. apt-get update

7. move to the folder name zeek with cd ./zeek

8. ./configure

9. make

10. make install

11. ./configure --builddir=../zeek-buildtools

12. For easy access later on:

    Sudo vim .bashrc

    Go to the bottom of the file and insert: alias zeek='/usr/local/zeek/bin/zeek'

13. check that zeek is configured right by typing: zeek -v

Part 5. Zeek manual.

Command	Manual discerption	More information
<file>	Zeek script file, or read stdin	
-a|--parse-only	exit immediately after parsing scripts	
-b|--bare-mode	don't load scripts from the base/ directory	
-d|--debug-script	activate Zeek script debugging	
-e|--exec <zeek code>	augment loaded scripts by given code	
-f|--filter <filter>	tcpdump filter	
-h|--help	command line help	
-i|--iface <interface>	read from given interface (only one allowed)	Used to specify a network interface.
-p|--prefix <prefix>	add given prefix to Zeek script file resolution	
-r|--readfile <readfile>	read from given tcpdump file (only one allowed, pass '-' as the filename to read from stdin)	Option that tells Zeek that is will be reading from an offline file.
-s|--rulefile <rulefile>	read rules from given file	
-t|--tracefile <tracefile>	activate execution tracing	
-u|--usage-issues	find variable usage issues and exit; use -uu for deeper/more expensive analysis	
-v|--version	print version and exit	
-w|--writefile <writefile>	write to given tcpdump file	Specify that we will be writhing to a new file.
-C|--no-checksums	ignore checksums	Disable checksums validation.
-D|--deterministic	initialize random seeds to zero	
-F|--force-dns	force DNS	
-G|--load-seeds <file>	load seeds from given file	
-H|--save-seeds <file>	save seeds to given file	
-I|--print-id <ID name>	print out given ID	
-N|--print-plugins	print available plugins and exit (-NN for verbose)	
-O|--optimize[=<option>]	enable script optimization (use -O help for options)	
-o|--optimize-only=<func>	enable script optimization only for the given function	
-P|--prime-dns	prime DNS	
-Q|--time	print execution time summary to stderr	
-S|--debug-rules	enable rule debugging	
-T|--re-level <level>	set 'RE_level' for rules	

-U | --status-file <file>	Record process status in file	
-W | --watchdog	activate watchdog timer	
-X | --zeekygen <cfgfile>	generate documentation based on config file	
--pseudo-realtime[=<speedup>]	enable pseudo-realtime for performance evaluation (default 1)	
-j | --jobs	enable supervisor mode	
--test	run unit tests ('--test -h' for help, only when compiling with ENABLE_ZEEK_UNIT_TESTS)	
$ZEEKPATH	file search path (.:/usr/local/zeek/share/zeek:/usr/local/zeek/share/zeek/policy:/usr/local/zeek/share/zeek/site)	
$ZEEK_PLUGIN_PATH	plugin search path (/usr/local/zeek/lib/zeek/plugins)	
$ZEEK_PLUGIN_ACTIVATE	plugins to always activate ()	
$ZEEK_PREFIXES	prefix list ()	
$ZEEK_DNS_FAKE	disable DNS lookups (off)	
$ZEEK_SEED_FILE	file to load seeds from (not set)	
$ZEEK_LOG_SUFFIX	ASCII log file extension (.log)	
$ZEEK_PROFILER_FILE	Output file for script execution statistics (not set)	
$ZEEK_DISABLE_ZEEKYGEN	Disable Zeekygen documentation support (not set)	
$ZEEK_DNS_RESOLVER	IPv4/IPv6 address of DNS resolver to use (not set, will use first IPv4 address from /etc/resolv.conf)	
$ZEEK_DEBUG_LOG_STDERR	Use stderr for debug logs generated via the -B flag	

Part 6. Zeek possible log files.

-Detection

Intel.log- Intelligence data matches.

Notice.log- Zeek notices.

Notice_alarm.log- The alarm stream.

Signatures.log- Signature matches.

Traceroute.log- Traceroute detection.

-Files

Files.log- File analysis results.

Ocsp.log- Online Certificate Status Protocol (OCSP), Only created if policy script is loaded.

Pe.log- Portable Executable (PE).

X509.log- X.509 certificate info.

-Miscellaneous

barnyard2.log-Alerts received from Barnyard2.

dpd.log- Dynamic protocol detection failures.

unified2.log- Interprets Snort's unified output.

unknown_protocols.log- Information about packet protocols that Zeek doesn't know how to process.

weird.log- Unexpected network-level activity.

weird_stats.log- Statistics about unexpected activity.

-Net Control

netcontrol.log- Netcontrol actions.

netcontrol_drop.log- Netcontrol actions.

netcontrol_shunt.log- Netcontrol shunt actions.

netcontrol_catch_release.log- Netcontrol catch and release actions.

openflow.log- OpenFlow debug log.

-Network Observations

known_certs.log- SSL (Secure Sockets Layer) certificates.

known_hosts.log- Hosts that have completed Transmission Control Protocol handshakes.

known_modbus.log- Modbus masters and slaves.

known_services.log- Services running on hosts.

software.log- Software being used on the network.

-Network Protocols

Conn.log- TCP/UDP/ICMP connections.

Dce_rpc.log- Distributed computing environment/ PRC.

Dhcp.log- Dynamic Host Configuration Protocol leases.

Dnp3.log- DNP3 requests and replies.

Dns.log- Domain Name System activity.

Ftp.log- File Transfer Protocol activity.

Http.log- Hypertext Transfer Protocol requests and replies.

Irc.log- Internet Relay Chat commands and responses.

Kerberos.log- Kerberos.

Modbus.log- Modbus commands and responses.

modbus_register_change.log- Tracks changes to Modbus holding registers.

mysql.log- MySQL.

Ntlm.log- NT LAN Manager (NTLM).

Ntp.log- Network Time Protocol.

radius.log- RADIUS authentication attempts.

rdp.log- Remote Desktop Protocol.

rfb.log- Remote Framebuffer (RFB).

sip.log- Session Initiation Protocol

smb_cmd.log- Server Message Block commands.

smb_files.log- Server Message Block files.

smb_mapping.log- Server Message Block trees.

Smtp.log- Simple Mail Transfer Protocol transactions.

Snmp.log- Simple Network Management Protocol messages.

socks.log- SOCKS proxy requests.

Ssh.log- SSH connections.

Ssl.log- SSL/TLS handshake info.

Syslog.log- Syslog massages.

Tunnel.log- Tunneling protocol events.

-Zeek Diagnostics

broker.log- Peering status events between Zeek or Broker- enabled processes.

capture_loss.log- Packet loss rate.

cluster.log- Zeek cluster messages.

config.log- Configuration option changes.

loaded_scripts.log- Shows all scripts loaded by Zeek.

packet_filter.log- List packet filters that were applied.

print.log- Print statements that were redirected to a log stream.

prof.log- Profiling statistics (to create this log, load policy/misc/profiling.zeek).

reporter.log- Internal error/ warning/ info messages.

stats.log- Memory/ event/ packet/ lag statistics.

stderr.log- Captures standard error when Zeek is started from ZeekControl.

stdout.log- Captures standard output when Zeek is started from ZeekControl.

Part 7. Addendum

https://docs.zeek.org/en/master/

https://docs.zeek.org/en/current/script-reference/log-files.html

https://docs.zeek.org/en/master/about.html

http://ce.sc.edu/cyberinfra/docs/workshop/Zeek_Lab_Series.pdf

https://cybersecurity.att.com/blogs/security-essentials/open-source-intrusion-detection-tools-a-quick-overview

By Sean Yarkoni