

ZEEK CHEAT SHEET

Network Security Monitor & Analysis Framework

■ BASIC COMMANDS

Command	Description
zeek -v	Display Zeek version
sudo su	Elevate privileges (required to run Zeek)
zeekctl	Start ZeekControl interactive module
zeekctl status	Check Zeek service status
zeekctl start	Start Zeek service
zeekctl stop	Stop Zeek service

■ PROCESSING PCAP FILES

Command	Description
zeek -C -r sample.pcap	Process pcap file (-C ignores checksum errors)
zeek -C -r file.pcap -s sig.sig	Process with signature file
zeek -C -r file.pcap script.zeek	Process with Zeek script
zeek -C -r file.pcap local	Run with all local scripts

■ Default logs path: /opt/zeek/logs/ • PCAP processing creates logs in current directory

■ NAVIGATING & READING LOGS

Command	Description
ls	List generated log files
cat conn.log	Display full contents of connection log
cat conn.log zeek-cut uid proto	Extract specific columns from log
cat conn.log zeek-cut id.orig_h id.resp_h	Extract source and destination IPs
cat notice.log zeek-cut msg	View notice messages
cat signatures.log zeek-cut event_msg	View signature match events
grep -rin "pattern" *	Search across all log files

■ Common zeek-cut fields:

uid • proto • id.orig_h • id.orig_p • id.resp_h • id.resp_p • signature • src_addr • dst_addr
• sig_id • event_msg • sub_msg

■ SIGNATURE CONDITIONS & FILTERS

Condition	Description
src-ip / dst-ip	Source / Destination IP address
src-port / dst-port	Source / Destination port
ip-proto	Protocol: TCP, UDP, ICMP, ICMP6, IP, IP6
payload	Packet payload content
http-request	Decoded HTTP requests
http-request-header	Client-side HTTP headers
http-request-body	Client-side HTTP request body
http-reply-header	Server-side HTTP headers
http-reply-body	Server-side HTTP response body
ftp	FTP session command line input
same-ip	Filter for duplicate src/dst addresses
event	Signature match message to generate

■ SIGNATURE FILE EXAMPLES

■ HTTP Password Detection (save as http-password.sig):

```
signature http-password {  
  ip-proto == tcp  
  dst_port == 80  
  payload /. *password.*/  
  event "Cleartext Password Found!"  
}
```

■ FTP Username Detection:

```
signature ftp-username {  
  ip-proto == tcp  
  ftp /. *USER.*/  
  event "FTP Username Input Found!"  
}
```

■ FTP Brute-force Detection:

```
signature ftp-brute {  
  ip-proto == tcp  
  payload /. *530.*Login.*incorrect.*/  
  event "FTP Brute-force Attempt!"  
}
```

■ ZEEK SCRIPTS

■ Important Paths:

Path	Description
/opt/zeek/share/zeek/base	Base scripts location
/opt/zeek/share/zeek/site	User-generated scripts
/opt/zeek/share/zeek/policy	Policy scripts
/opt/zeek/share/zeek/site/local.zeek	Main configuration file
/opt/zeek/share/zeek/base/bif	Built-in functions
/opt/zeek/share/zeek/base/protocols	Protocol analyzers

■ Basic Script Template (101.zeek):

```
event zeek_init() {
    print ("Started Zeek!");
}

event zeek_done() {
    print ("Stopped Zeek!");
}
```

■ Connection Monitor Script:

```
event new_connection(c: connection) {
    print fmt("Source: %s:%s", c$id$orig_h, c$id$orig_p);
    print fmt("Dest: %s:%s", c$id$resp_h, c$id$resp_p);
}
```

■ Signature Match Handler:

```
event signature_match(state: signature_state, msg: string, data: string) {
    if (state$sig_id == "ftp-admin") {
        print ("Signature hit! → #FTP-Admin");
    }
}
```

■ DHCP Hostname Extractor:

```
event dhcp_message(c: connection, is_orig: bool, msg: DHCP::Msg, options: DHCP::Options) {
    print options$host_name;
}
```

■ FRAMEWORKS

■ File Hashing Framework:

```
@load /opt/zeek/share/zeek/policy/frameworks/files/hash-all-files.zeek
```

```
cat files.log | zeek-cut md5 sha1 sha256
```

■ File Extraction Framework:

```
zeek -C -r case.pcap /opt/zeek/share/zeek/policy/frameworks/files/extract-all-files.zeek  
ls extract_files | nl # List extracted files
```

■ Intelligence Framework (intel-demo.zeek):

```
@load policy/frameworks/intel/seen  
@load policy/frameworks/intel/do_notice  
redef Intel::read_files += { "/opt/zeek/intel/zeek_intel.txt" };
```

■ Built-in Brute Force Detection:

```
zeek -C -r ftp.pcap /opt/zeek/share/zeek/policy/protocols/ftp/detect-bruteforcing.zeek
```

■ PACKAGE MANAGER (zkg)

Command	Description
zkg install package_path	Install package from path
zkg install git_url	Install from Git URL
zkg list	List installed packages
zkg remove package_name	Remove installed package
zkg refresh	Check for version updates
zkg upgrade	Update all installed packages

■ Useful Packages:

Package	Usage
zeek-sniffpass	Detect cleartext passwords
geopip-conn	Add geolocation to conn.log

```
zkg install zeek/cybera/zeek-sniffpass  
zeek -Cr http.pcap zeek-sniffpass
```

■ USEFUL LOG ANALYSIS COMMANDS

Command	Purpose
cat signatures.log zeek-cut src_addr dst_addr event_msg sub_msg sort -r uniq	Unique signature matches
cat notice.log zeek-cut id.orig_h id.resp_h proto note msg	Notice details with IPs
cat files.log zeek-cut fuid conn_uids tx_hosts rx_hosts mime_type extracted nl	File transfer details

<code>cat conn.log zeek-cut geo.orig.country_code geo.resp.city</code>	GeoIP info (with <code>geoip-conn</code>)
<code>grep -rin FILE_ID * column -t nl less -S</code>	Track file across all logs

■ QUICK REFERENCE

Task	Command
Process PCAP	<code>zeek -C -r file.pcap</code>
With signature	<code>zeek -C -r file.pcap -s signature.sig</code>
With script	<code>zeek -C -r file.pcap script.zeek</code>
All local scripts	<code>zeek -C -r file.pcap local</code>
Extract columns	<code>cat log.log zeek-cut field1 field2</code>
Hash all files	<code>zeek -C -r file.pcap hash-all-files.zeek</code>
Extract files	<code>zeek -C -r file.pcap extract-all-files.zeek</code>

Zeek Documentation: <https://docs.zeek.org> • Package Index: <https://packages.zeek.org>