Naqash Abbas

Cybersecurity analysts | Network Traffic Analyst | Open-Source Advocate

- ◆ LinkedIn Profile: https://www.linkedin.com/in/nagash-abbas-314661282/
- **♦ Location:** Pakistan
- ◆ Lab Environment: Kali Linux (VM), Zeek Network Security Monitor
- ◆ Tool Focus: Zeek 8.x (Built from Source)

About This Manual

This hands-on lab manual is the result of my independent exploration and practical implementation of **Zeek**, a powerful open-source network monitoring framework. Every step from installation to traffic analysis, was tested on my Kali Linux machine, and each section includes:

- Real command outputs
- Annotated screenshots
- Q Log file analysis
- Custom notes for understanding

This document not only showcases my technical capability, but also my attention to detail, documentation skills, and practical grasp of network-level threat detection.

— Naqash Abbas

[&]quot;Security is not a product, but a process and mastering tools like Zeek puts us in control of that process."

V ZEEK PROFESSIONAL MANUAL

■ INTRODUCTION TO ZEEK

What is Zeek?

Zeek is an open-source network security monitor that passively observes network traffic and generates high-level logs and events. It's used for:

- Detecting intrusions
- Monitoring policy violations
- Analyzing network behavior
- · Logging encrypted and unencrypted traffic details

Why Use Zeek?

- Works passively on network taps or SPAN ports
- Highly customizable with its scripting language
- Used in SOCs, threat hunting, and forensic analysis

SYSTEM REQUIREMENTS

- OS: Linux (Ubuntu/Kali/CentOS), FreeBSD, or macOS
- RAM: Minimum 2 GB (8 GB recommended for large traffic)
- Tools: libpcap, cmake, make, gcc/g++, Python 3

Step-by-Step Installation of Zeek on Kali Linux

♦ Step 1: Update System & Install Dependencies

Make sure your system is up to date and install required packages:

```
sudo apt update && sudo apt upgrade -y
sudo apt install cmake make gcc g++ flex bison \
libpcap-dev libssl-dev python3-dev swig zlib1g-dev \
libcurl4-openssl-dev libmaxminddb-dev libgeoip-dev \
liblzma-dev libmagic-dev libjemalloc-dev git -y
```

These are essential development libraries needed to build Zeek from source.

```
Libpcap-dev Libsci-dev python3-dev swig zlibig-dev \
Libcra-dev Libsci-dev python3-dev swig zlibig-dev \
Libcra-dev Libsci-dev python3-dev swig zlibig-dev \
Libcra-dev Libsci-dev libsealtoc-dev sig zlibig-dev \
Libcra-dev Libsci-dev libsealtoc-dev sig zlibig-dev \
Libcra-dev Libsci-dev Libsci-dev libsealtoc-dev git -y
Hill: https://packages.wan/con/e.x/aya table InRelease (41.5 kB]
Get:3 http://kali.domolnod/kali kait-rolling/main and64 Contents (640) [51.4 kB]
Hill: https://pep.omogodb.org/apt/debian/bookworm/mongodb-org/7.0 InRelease
Get:2 http://kali.domolnod/kali kait-rolling/main and64 Contents (640) [51.4 kB]

ZB packages can be upgraded. Bun 'apt List -upgradable' to see them.
marrings https://repo.omogodb.org/apt/debian/distr/bookworm/mongodb-org/7.0/InRelease: Policy will reject signature within a year, see --audit for details
Upgrading:
chrosium:
comsole-setup firefox.esr keyboard-configuration libsebconfelient0
Libscium-comsole-setup-linux freetds-common libapache2-mod-phps.4 Libs/yulibre-text Libscium-3.0-0-0 pperl phps.4-ppcache

Summary:
Upgrading: 28, Installing: 0, Removing: 0, Not Upgrading: 0
Download size: 185 MB
Space needed: 89.1 kB / 10.1 GB available

Get:1 http://kali.domolnod/kali kait-rolling/main and64 perl-modules-5.40 and64 5.40.1-5 [4,335 kB]
Get:2 http://kali.domolnod/kali kait-rolling/main and64 perl-modules-5.40 all 5.40.1-5 [3,02 kB]
Get:3 http://kali.domolnod/kali kait-rolling/main and64 perl-modules-5.40 all 5.40.1-5 [3,02 kB]
Get:4 http://kali.domolnod/kali kait-rolling/main and64 perl-modules-5.40 all 5.40.1-5 [3,02 kB]
Get:5 http://kali.domolnod/kali kait-rolling/main and64 perl-modules-5.40 all 5.40.1-5 [3,02 kB]
Get:6 http://kali.domolnod/kali kait-rolling/main and64 perl-modules-5.40 all 5.40.1-5 [3,02 kB]
Get:6 http://kali.domolnod/kali kait-rolling/main and64 perl-modules-5.40 all 5.40.1-5 [3,02 kB]
Get:16 http://kali.domolnod/kali kait-rolling/main and64 chromium-common and64 138.0.7244.160-1 [10.7 kB]
Get:16 http://kali.domolnod/kali kait-rolling/main and64 chromium
```

Step 2: Clone the Zeek Source Code

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

cd zeek

The --recursive flag ensures all submodules (like zkg, the Zeek Package Manager) are cloned too.

```
Les et clone —recursive https://github.com/reck/reck
cd zeck

Cloning into 'zeck'...
remote: Enumerating objects: 255975, done.
remote: Counting objects: 108% (1732/1732) done.
remote: Counting objects: 108% (1732/1732) done.
remote: Counting objects: 108% (1732/1732) done.
remote: Total 255075 (delta 1264), peused 399 (doita 964), pack-reused 253343 (from 3)
Receiving objects: 108% (1953/1816310), done.
Submodule' auxil/bafc! (Intps://github.com/zeck/binpac) registered for path 'auxil/binpac'
Submodule' auxil/bafc! (Intps://github.com/zeck/binpac) registered for path 'auxil/binpac'
Submodule' auxil/broker (Intps://github.com/zeck/binpac) registered for path 'auxil/binpac'
Submodule' auxil/broker (Intps://github.com/zeck/seck-doitage)
Submodule' auxil/penser (Intps://github.com/zeck/seck-doitage)
Submodule' auxil/penser (Intps://github.com/zeck/gen-zam) registered for path 'auxil/cares'
Submodule' auxil/penser (Intps://github.com/zeck/gen-zam) registered for path 'auxil/gen-zam'
Submodule' auxil/penser (Intps://github.com/zeck/gen-zam) registered for path 'auxil/lockquene'
Submodule' auxil/penser (Intps://github.com/zeck/gen-recorrector) registered for path 'auxil/lockquene'
Submodule' auxil/penser (Intps://github.com/zeck/pack-recorrector) registered for path 'auxil/penser (Intps://github.com/zeck/pack/packgem-annager) registered for path 'auxil/penser (Intps://github.com/zeck/pack/packgem-annager) registered for path 'auxil/penser (Intps://github.com/zeck/pack/packgem-annager) registered for path 'auxil/penser (Intps://github.com/zeck/pack-recorrector) registered for path 'auxil/penser (Intps://github.com/zeck/packgen-annager) registered for path 'auxil/penser (Intps://github.com/zeck/packgen-annager) registered for path 'auxil/pens
```

♦ Step 3: Build Configuration

Create a separate directory for the build process:

mkdir build

cd build

../configure

The configure script checks for dependencies and prepares the Makefile.

```
L-5 moder build

actionfigure

Using cmake version 3.31.6

Build Directory: /bowe/nagas//zeek

- The C compiler identification is GNU 14-2.0

- The CXX compiler identification is GNU 14-2.0

- The CXX compiler identification is GNU 14-2.0

- The CXX compiler identification is GNU 14-2.0

- Detecting C compiler ABI info - done
- Check for working C compiler ABI info - done
- Check for working C compiler identification
- Detecting CXX compiler identification
- Detecting CXX compiler identifier
- Detecting CXX compiler features
- done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Detecting CXX compiler features - done
- Dete
```

Step 4: Compile Zeek

make -j\$(nproc)

The -j\$(nproc) option uses all available CPU cores to speed up the build.

This step can take several minutes depending on your system specs.

Step 5: Install Zeek

sudo make install

This will install Zeek binaries to /usr/local/zeek.

```
Les and password for magash:

(W) Built target bif-()

(W) Built target
```

Step 6: Add Zeek to System PATH

To use Zeek from anywhere in the terminal, add it to your PATH:

export PATH=/usr/local/zeek/bin:\$PATH

You also ensured Bash is your default shell with:

chsh -s /bin/bash

Step 7: Verify Installation

zeek --version

You should see an output like:

zeek version 6.0.x

```
(naqash⊕ kali)-[~/zeek/build]

$\frac{1}{2}$ zeek --version

$\text{zeek version 8.0.0-dev.815}$
```

Step 8: Test Zeek with a PCAP File

Test Zeek with sample network traffic:

Wget https://raw.githubusercontent.com/zeek/zeek/master/testing/btest/Traces/http/get.trace -O test.pcap

zeek -r test.pcap

- Zeek will process the file and automatically create log files in the current folder, such as:
 - conn.log → Connection logs
 - http.log → HTTP request details
 - weird.log → Unexpected protocol behavior
 - notice.log → Any triggered alerts (if present)

♦ 2. View the Output Logs

You can now inspect the logs. Try:

```
(naqash⊗ kali)-[~/zeek/build]
$\frac{1}{2} \text{ls *.log}
conn.log files.log http.log packet_filter.log
```

Then view individual logs:

cat conn.log

cat http.log

Or more readable:

less conn.log

less http.log

You can also extract specific fields using zeek-cut:

zeek-cut id.orig_h id.resp_h < conn.log
zeek-cut host uri < http.log</pre>

press q to exit

```
Commands marked with * may be preceded by a number, N.
Notes in parentheses indicate the behavior if N is given.
A key preceded by a caret indicates the Ctrl key; thus 'K is ctrl-K.

h H Display this help.
q :q Q :Q ZZ Exit.

**NOVING**

**Noving
```

Inspect More Logs

cat conn.log

cat dns.log

cat weird.log

cat notice.log

```
L-$ cat conn.log
cat dns.log
cat dns.log
cat notice.log
Hsparator Nu9
Hsparator Nu9
Hsparator Signarator
Humber_field (empty)
Humber_field (empty)
Humber_field = 000
Hopen 2025-08-01-02-57-25
Higher to sid id.orig_b id.orig_p id.resp_b id.resp_b proto service duration orig_bytes resp_bytes conn_state local_orig local
l_resp missed_bytes history orig_pkts orig_ip_bytes resp_pkts resp_ip_bytes tunnel_parents ip_proto
Htypes time string addr port addr port enum string interval count count string bool bool count string count count count set[string]count
ount
1026092250.8609344 Cseshx2lqHQAp70VG8 141.142.228.5 59856 192.150.187.43 80 tcp http 0.211484 136 5007 SF F F 0 ShADadFF 7
Files 2025-08-01-02-57-25
cat: dns.log: No such file or directory
cat: notice.log: No such file or directory
```

Analyze Specific Traffic

Use filtering tools like grep:

grep "192.150" http.log





Bonus: Zeek Package Manager (ZKG)

ZKG lets you install Zeek scripts from the community:

zkg list

zkg install zeek/zeek-af-packet-plugin

Final Thoughts

You've now set up **Zeek** from source on **Kali Linux** — a powerful tool in any security analyst's arsenal.

Zeek can be extended or integrated with:

- Elasticsearch & Kibana
- Wazuh / SIEM
- Suricata / IDS
- Threat intelligence feeds
- **Custom detection scripts**