runZero

# Shaking Out Shells With SSHamble


HD MOORE | AUGUST 9, 2025

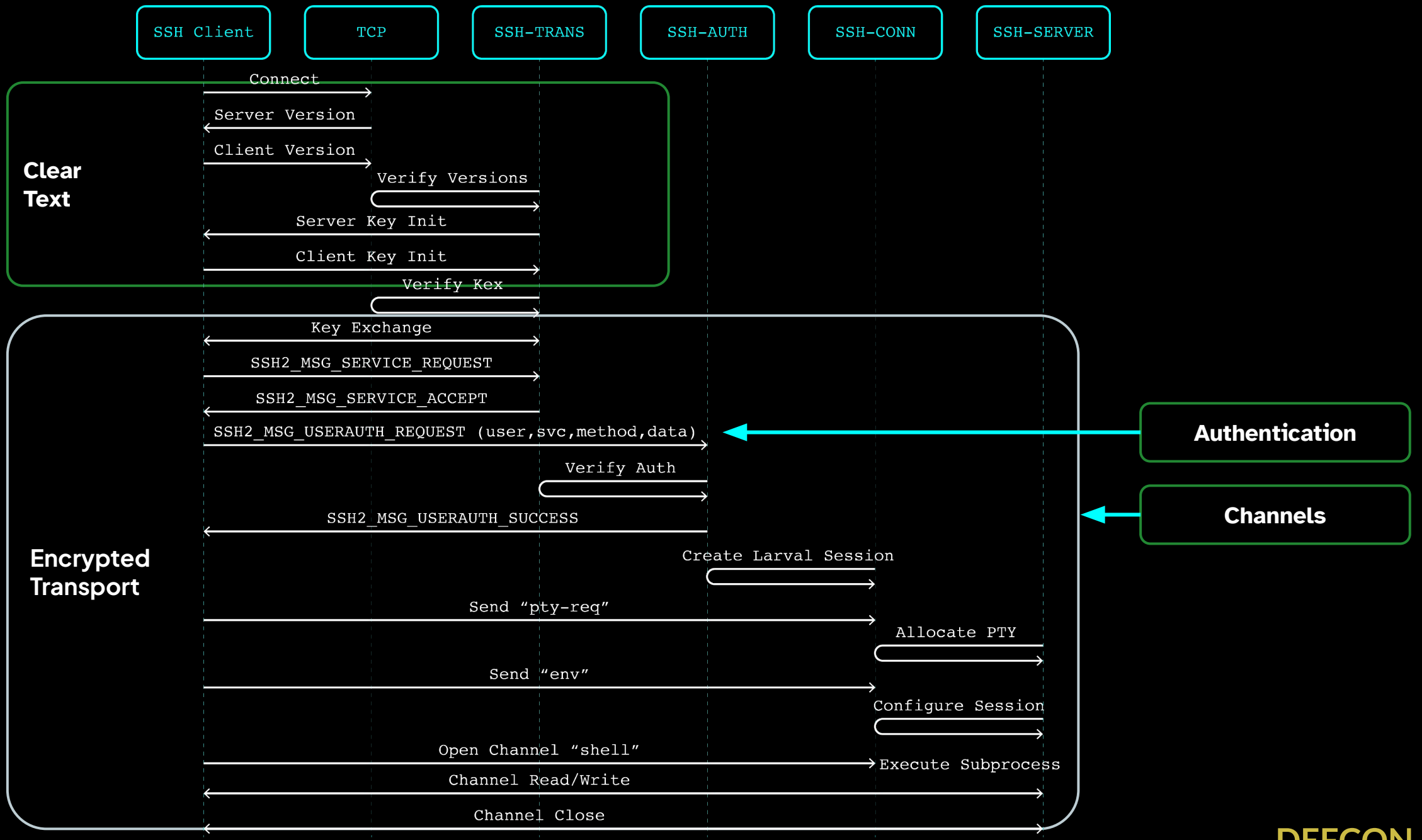
with contributions from Rob King

# Agenda

**A 20-minute follow-up & extension of our DC 32 research[1]**

- → A fast overview of the SSH protocol and ecosystem
- → A recap of major SSH exposures since last year
- → New research, vulnerabilities, and exposure stats
- → Updates to our open source tooling!

1. https://www.runzero.com/blog/sshamble-unexpected-exposures-in-the-secure-shell/


SSH Client
TCP
SSH-TRANS
SSH-AUTH
SSH-CONN
SSH-SERVER
Clear Text
Connect
Server Version
Client Version
Verify Versions
Server Key Init
Client Key Init
Verify Kex
Encrypted Transport
Key Exchange
SSH2_MSG_SERVICE_REQUEST
SSH2_MSG_SERVICE_ACCEPT
SSH2_MSG_USERAUTH_REQUEST (user,svc,method,data)
Verify Auth
SSH2_MSG_USERAUTH_SUCCESS
Create Larval Session
Send "pty-req"
Allocate PTY
Send "env"
Configure Session
Open Channel "shell"
Execute Subprocess
Channel Read/Write
Channel Close
Authentication
Channels

# SSH pre-authentication information exposure

**TCP/IP** → **Server Version** → **Kex Init Extensions** → **Server Banner** → **Authentication**

TCP window size & scaling factors can determine the OS & kernel versions.

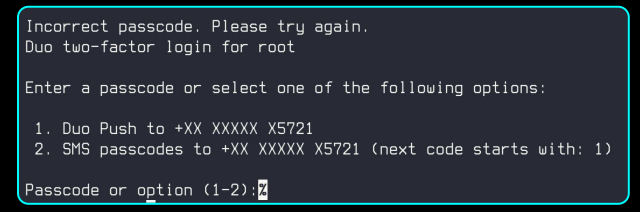
Protocol version, implementation, & package version.

Ciphers, MACs, key exchange protocols, compression methods, & server-side extensions.

Pre-authentication "banner" can be extensive, especially with network equipment.

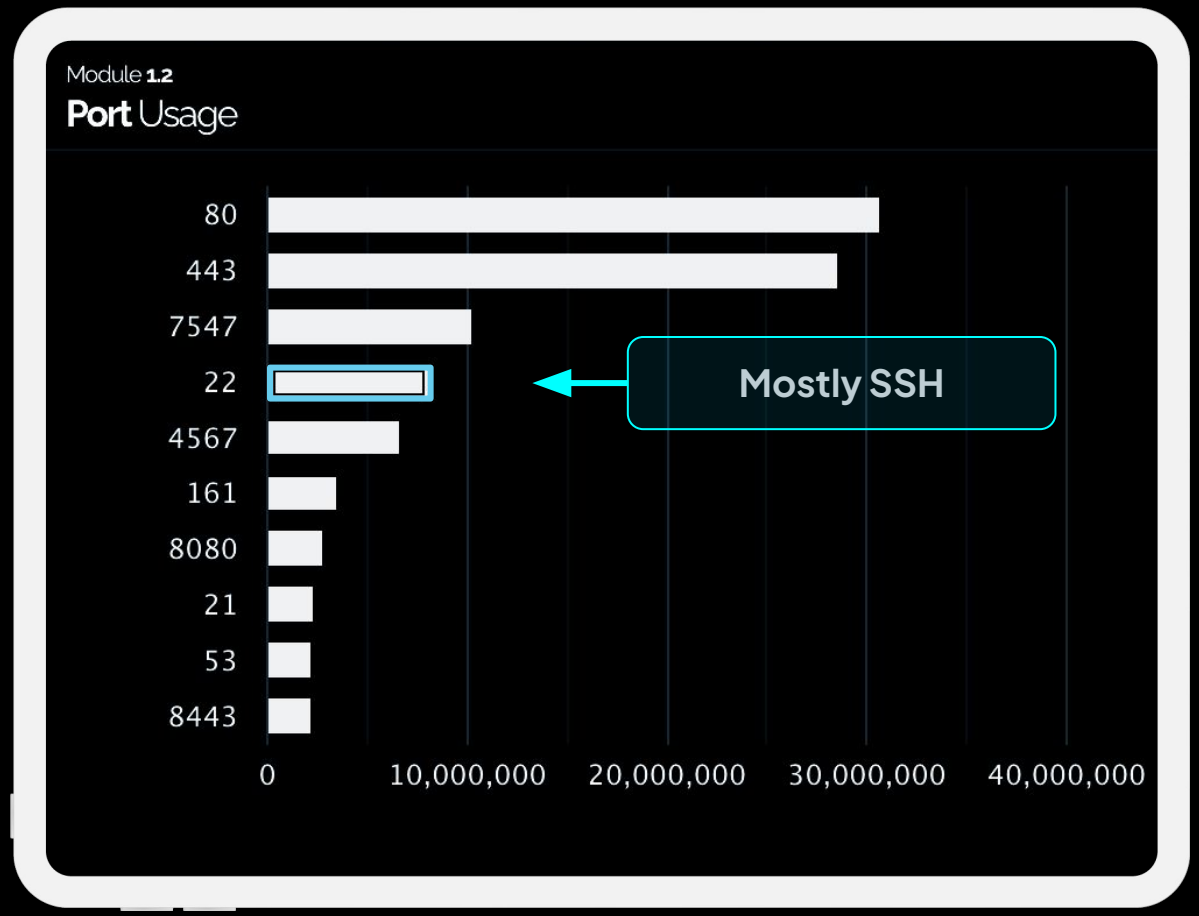
Authentication method list, public key testing, failed auth limits, & interactive questions & prompts.

| Platform | Version | SSH banner | v4 tcp.win | v4 MSS | MSS Multiplier | v4 Window Scale |
|---|---|---|---|---|---|---|
| CentOS Linux | 7.1 | SSH-2.0-OpenSSH_6.6.1 | 14480 | 1460 | 10 | 7 |
| CentOS Linux | 7.2 | SSH-2.0-OpenSSH_6.6.1 | 28960 | 1460 | 20 | 7 |
| CentOS Linux | 7.3 | | 28960 | 1460 | 20 | 7 |
| CentOS Linux | 7.4 | SSH-2.0-OpenSSH_7.4 | 28960 | 1460 | 20 | 7 |
| CentOS Linux | 7.5 | | 28960 | 1460 | 20 | 7 |
| Oracle Linux Server | 7.7 | | 28960 | 1460 | 20 | 7 |
| CentOS Linux | 7.9 | | 28960 | 1460 | 20 | 7 |
| Oracle Linux Server | 7.9 | | 28960 | 1460 | 20 | 7 |
| Scientific Linux | 7.9 | | 28960 | 1460 | 20 | 7 |
| CentOS Linux | 8.0 | SSH-2.0-OpenSSH_7.8 | 28960 | 1460 | 20 | 7 |
| Oracle Linux Server | 8.0 | | 28960 | 1460 | 20 | 7 |

# SSH is everywhere

- Second-most common remote admin service behind HTTP
- Enabled by default in clouds
- Part of every major OS
- Embedded & servers
- Even mobile!




SHODAN https://exposure.shodan.io/#/US

# SSH is mostly* OpenSSH & Dropbear

| | |
|---|---|
| OpenSSH | 14,876,142 |
| Dropbear sshd | 678,520 |
| Cisco IOS | 148,007 |
| Mikrotik | 125,545 |
| Linksys WRT45G modified dropbear sshd | 34,694 |
| lancom sshd | 29,559 |
| HP Integrated Lights-Out mpSSH | 6,145 |
| SCS sshd | 6,085 |
| ZyXEL ZyWALL sshd | 5,293 |
| WeOnlyDo sshd | 4,384 |
| DrayTek Vigor 2820n ADSL router sshd | 1,462 |
| Cisco/3Com IPSSHd | 1,388 |

## Not-OpenSSH/Dropbear are important

**Firewall, networking, & storage**

→ Cisco, NetScreen, Adtran, ComWare, Lancom

**OT/ICS equipment**

→ Siemens, NetPower, Mocana, CradlePoint, Digi

**Sensitive applications**

→ MOVEIT, CrushFTP, GlobalScape, JSCAPE

→ BitVis, GoAnywhere, ConfD (Erlang)

→ Gerrit, Forgejo, Gitlab

SHODAN

# SSH provides transport & authentication

**Version exchange & kex init in the clear**

- → Version: SSH-2.0 OpenSSH-9.8p1 deb13u3
- → Ciphers, MACs, Compressions, Languages, etc

**Key exchange to negotiate secure transport**

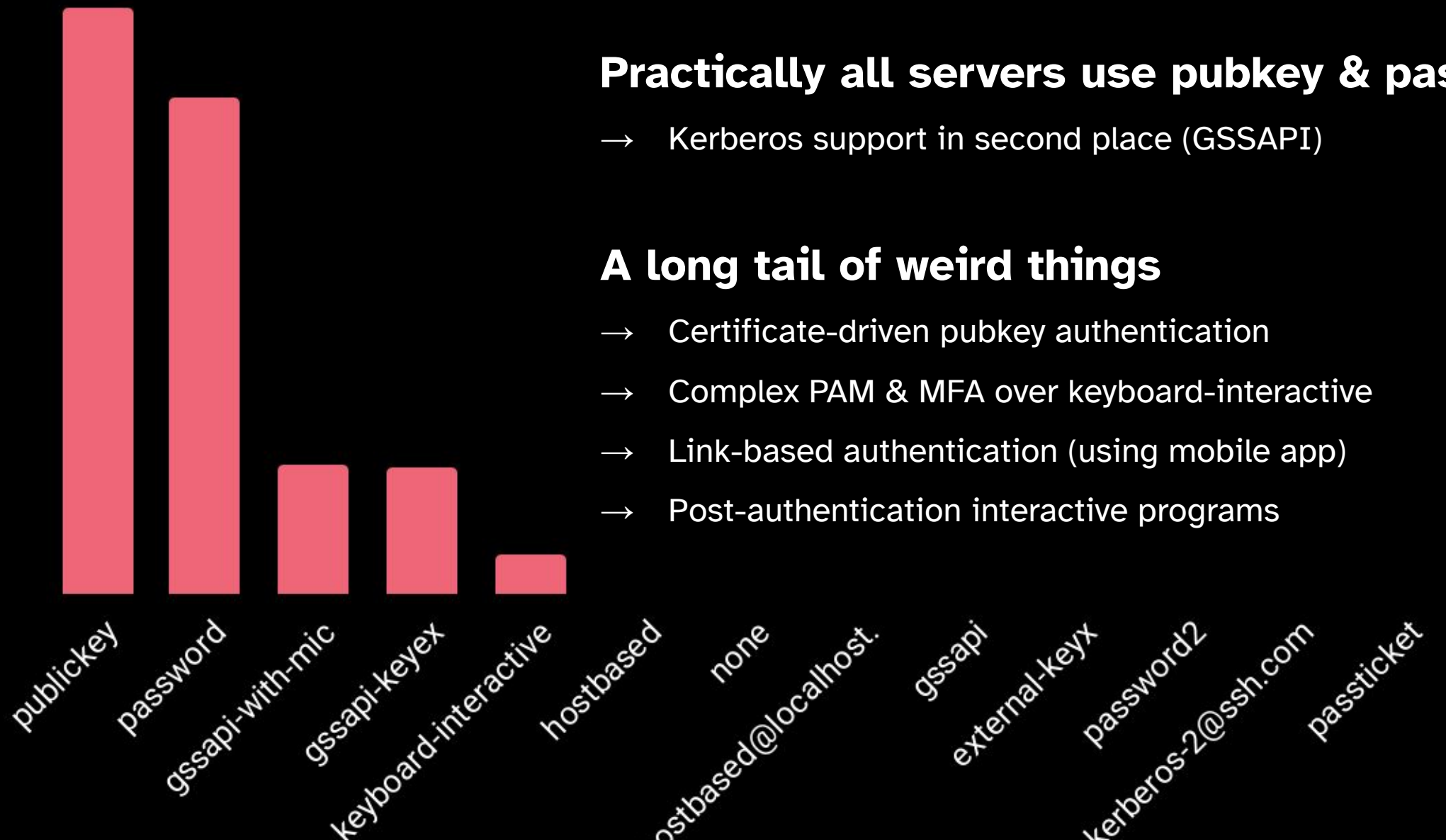
- → Diffie-Hellman & friends pinned with server host key(s)
- → Algorithm picked by kex init agreement

**Authentication using one or more methods**

- → Passwords, public keys, kerberos, & more
- → PK uses the session ID for proof signing

Similar to TLS

# SSH authentication




## Practically all servers use pubkey & password

→ Kerberos support in second place (GSSAPI)

## A long tail of weird things

→ Certificate-driven pubkey authentication

→ Complex PAM & MFA over keyboard-interactive

→ Link-based authentication (using mobile app)

→ Post-authentication interactive programs

# Pubkey enables pre-auth user & key confirmation

## Servers

**A list of IP addresses or hostnames running SSH.**

**Scanners**

- nmap
- zmap
- masscan

**Databases**

- Shodan
- Censys
- Fofa.info

## Public Keys

**A list of public keys possibly linked to the target.**




## Usernames

**A list of usernames likely used by the target.**

**Defaults**

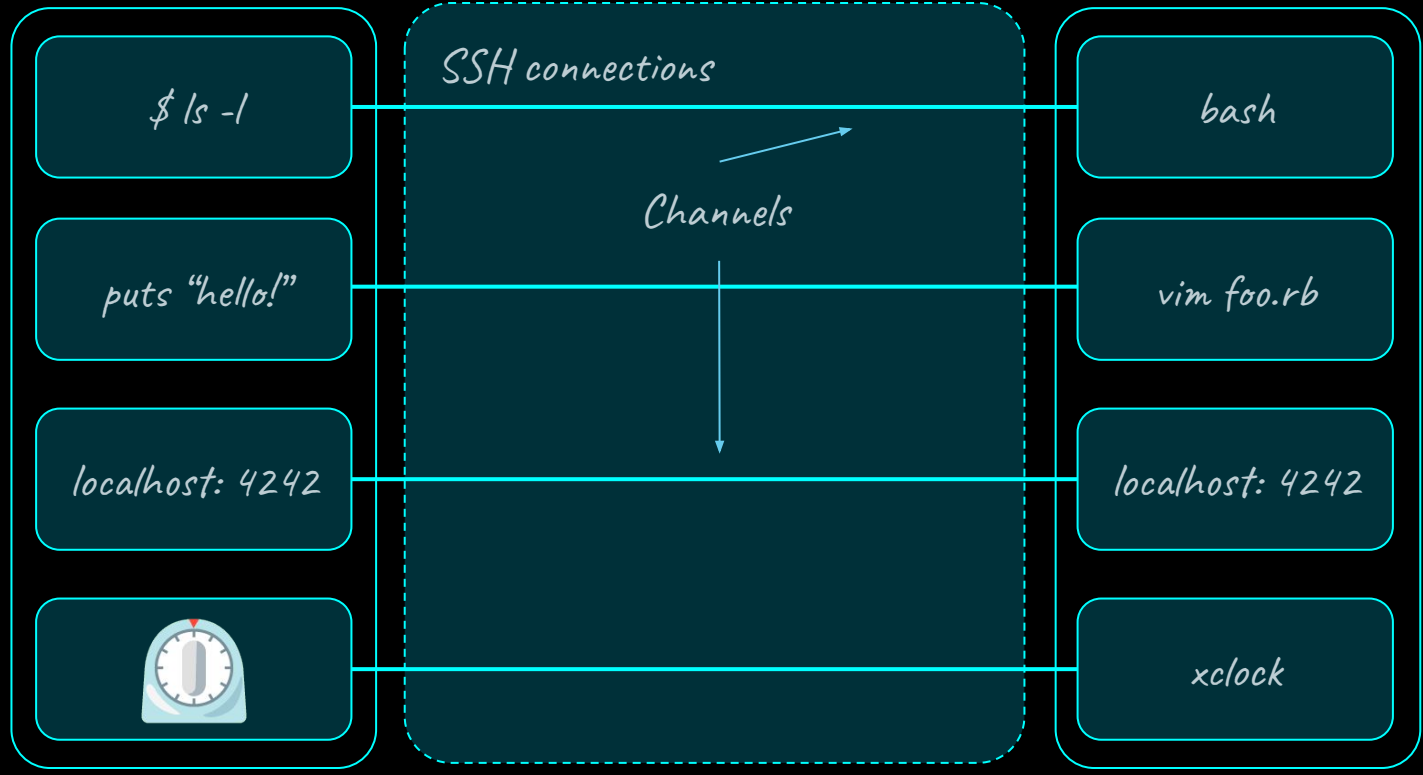
- root
- ec2-user
- ubuntu

**Specific**

- Public key “comments”
- Common handles
- Email prefixes

# SSH post-authentication is multiplexed

- → Interactive shells
- → Command execution
- → File transfer (SCP, SFTP)
- → TCP forwarding
- → Unix socket forwarding
- → X11 display forwarding
- → Agent forwarding



# SSH is effectively the other secure transport

## An alternative to TLS, but not exactly the same

- → Server key management can be, but usually isn't CA-based
- → Authentication is a core stage of the protocol
- → Multiplexer & session commands are unique
- → SSH uses the first algorithm sent by the client & supported by the server

## Compliance schemes gloss over SSH

- → Vendors point to strong cipher/mac + authentication similar to TLS
- → SSH specifics are often missing, assume best practices
- → Key management is the biggest gap

# Recent Vulnerabilities & Exposures

# Terrapin Attack

## Breaking SSH Channel Integrity by Sequence Number Manipulation

**Fabian Bäumer**
Research Assistant, Ruhr University Bochum

CVE-2023-48795

# XZ Utils backdoor

## A multi-year campaign started in 2021 and triggered in 2024

→ “Jia Tan” persona was likely the product of a state actor

→ Nearly-perfect Nobody-But-Us backdoor in SSH

→ Backdoor targeted SSH via systemd patches

→ Limited to Debian/RHEL-based distros

## Caught at the last possible moment by Andres Freund

→ Noticed that sshd was using more CPU than it should

→ Backdoor made it into rolling releases only

**CVE-2024-3094**

# RegreSSHion

## Incredible work by the Qualys Threat Research Unit

→ Regression of a signal re-entrance vulnerability

→ Unauthenticated remote root code execution

→ Tough to exploit due to ASLR & timing

**CVE-2024-6387**

## Related issue discovered by Solar Designer

→ Specific to Red Hat builds of OpenSSH

→ Limited to the non-root privsep user

**CVE-2024-6409**

The patch was hidden in the PerSourcePenalties feature, released a month prior to the disclosure.

# MOVEit & IPWorks SSH

## Another MOVEit vulnerability, but this time in SSH

→ watchTowr Labs reversed the MOVEit patch for CVE-2024-3094

→ The attacker's unauthenticated public key blob is opened as a file

→ File path supports UNC and was used for authentication

→ Root cause was the third-party IPWorks library

→ Threaded a dozen needles to bypass auth


**CVE-2024-5806**

# OpenSSH MiTM & DoS

## More amazing work by the Qualys Threat Research Unit

→ Successful machine-in-the-middle (MitM) against OpenSSH clients

→ Abuses  VerifyHostKeyDNS error handling with memory exhaustion

→ Pre-auth denial of service via “ping” messages

**CVE-2025-26465**

**CVE-2025-26466**

# Go SSH Authentication Bypass

## Platform.sh team identified a footgun in Go's x/crypto/ssh

→ Public key handler is called for each key presented by the attacker

→ Buggy applications can use the wrong key for authentication

→ Best documented case is the NetApp Telegraf Agent

→ Footgun partially fixed via Go x/crypto/ssh update

**CVE-2024-45337**

# Cisco Unified CM hardcoded root password

## It’s 2025 and backdoor creds still happen

→ A development slip-up that affected a narrow set of versions (15.0.1.13010-1 to 15.0.1.13017-1)

→ A great example of how DenyUser or PublicKey-only authentication could help


**CVE-2025-20309**

# Erlang OTP SSH Remote Code Execution

**Fabian Bäumer, Marcus Brinkmann, Marcel Maehren, & Jörg Schwenk (Ruhr University Bochum)**

**CVE-2025-32433**

- → State machine bug, the fix limits acceptable message types by session state
- → Exploitable after the version and kex init, even before encryption starts, easy one-liner exploit
- → Direct remote evaluation of Erlang code

# SSHamble

- → A research tool for SSH implementations
- → Quickly scans and gathers detailed data
- → Interesting attacks against authentication
- → Post-session authentication attacks
- → Pre-authentication state transitions
- → Post-session enumeration
- → Easy timing analysis


https://SSHamble.com

# Erlang OTP SSH Remote Code Execution

## Why did we miss this with SSHamble?

→ Erlang doesn't reply to the channel open or exec in this state, causing SSHamble to timeout. Unfortunately neither do a lot of non-vulnerable things, so tests have to be Erlang/ConfD specific.

**CVE-2025-32433**

## Real-world impact

→ Few instances of Erlang-SSHD in the wild

→ Cisco NETCONF ConfD is based on Erlang

→ Direct RCE on Cisco NSO / ConfD systems

→ Not port 22, check 830, 2022, & 2024

→ Was left unpatched for over a month

→ Patch it yourself with `ssh:stop().`

*23:00:38.907100 <0.106.0> Server Channel info returned: {noreply,"#state{}"}*

# Recap of IPv4 exposure from August 2024

**A lot of broken SSH on the internet**

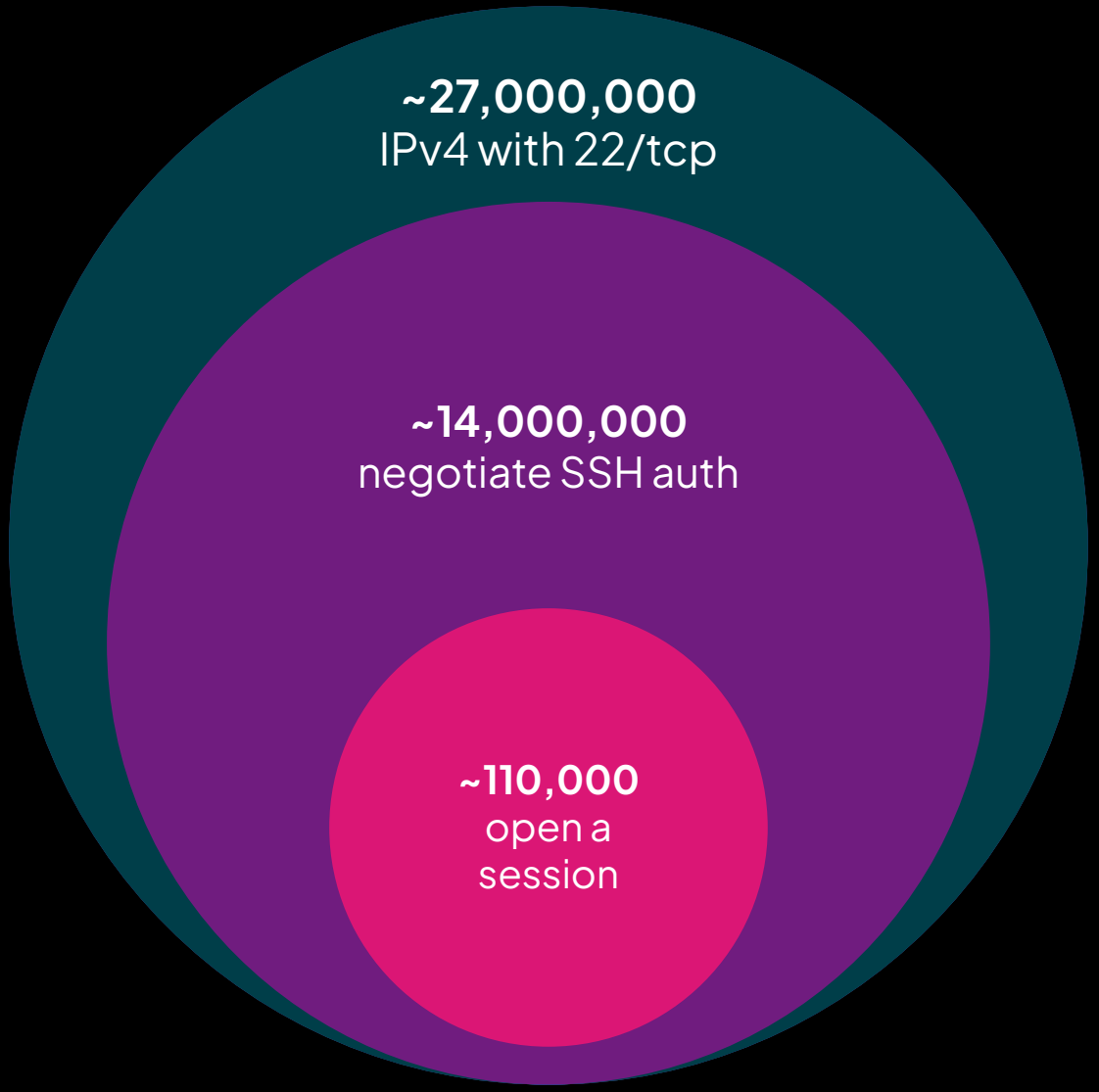
→ Tons of tarpits & buggy systems

→ ~14 million reach ssh-auth state

→ ~110k resulted in a session

→ ~9 unique vulnerabilities

**Scope limited to port 22**



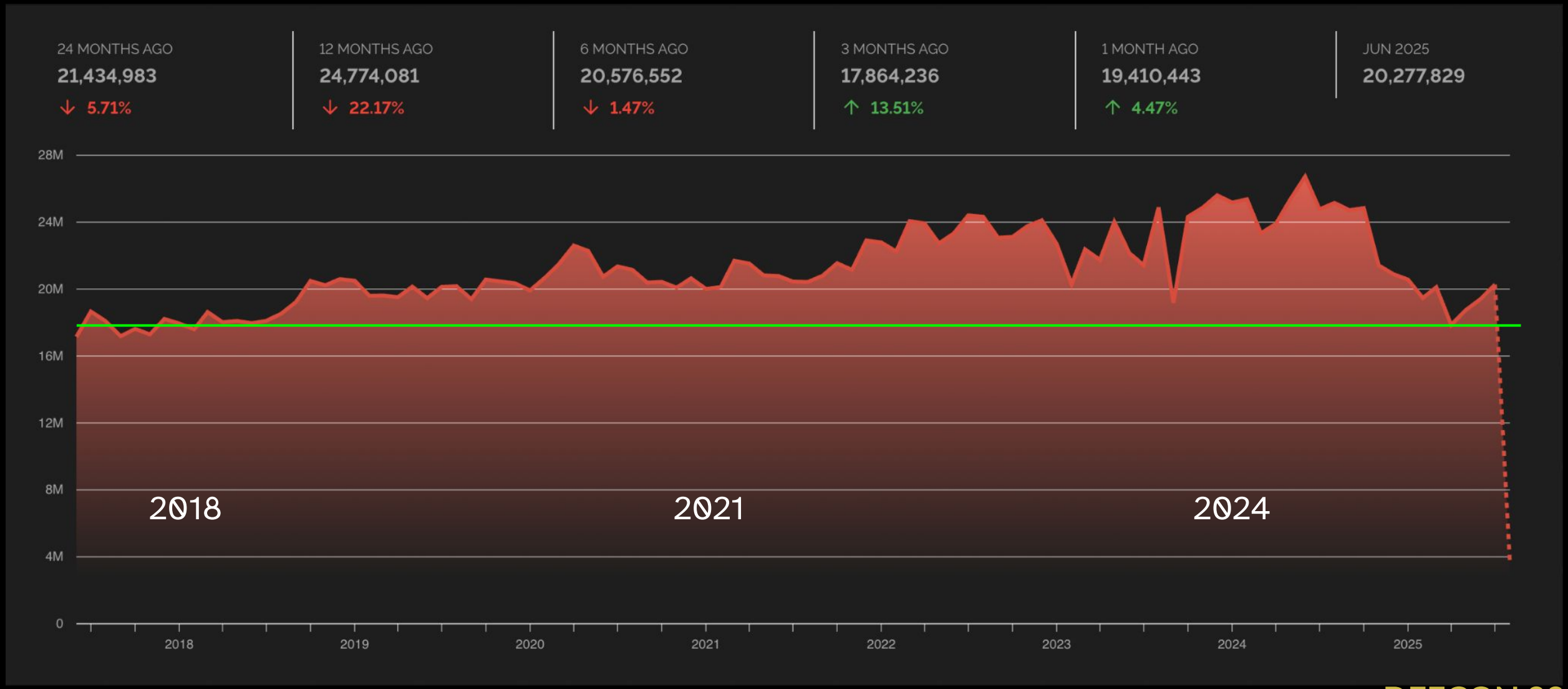
# SSHamble trophy case (2024)

| Product | Impact |
|---|---|
| Ruckus Wireless APs | Unauthenticated root command execution |
| Digi TransPort Gateways | Unauthenticated remote CLI access as SUPER |
| Panasonic Ethernet Switches | Unauthenticated remote CLI access as admin |
| Realtek ADSL Gateways | Unauthenticated remote CLI access as admin |
| Soft Serve | Authenticated remote code execution |
| GOGS | Authenticated remote command execution |
| OpenSSH for Windows | Unauthenticated OOB memory leak / comparison bug |
| ION Networks Service AP | Unauthenticated TCP forwarding |
| Multiple Products | Unlimited public key testing |

# 12 Months Later

# Total SSH exposure is flat since 2018

SHODAN



# Low uptake of PerSourcePenalties

## OpenSSH 9.8 added default rate limiting

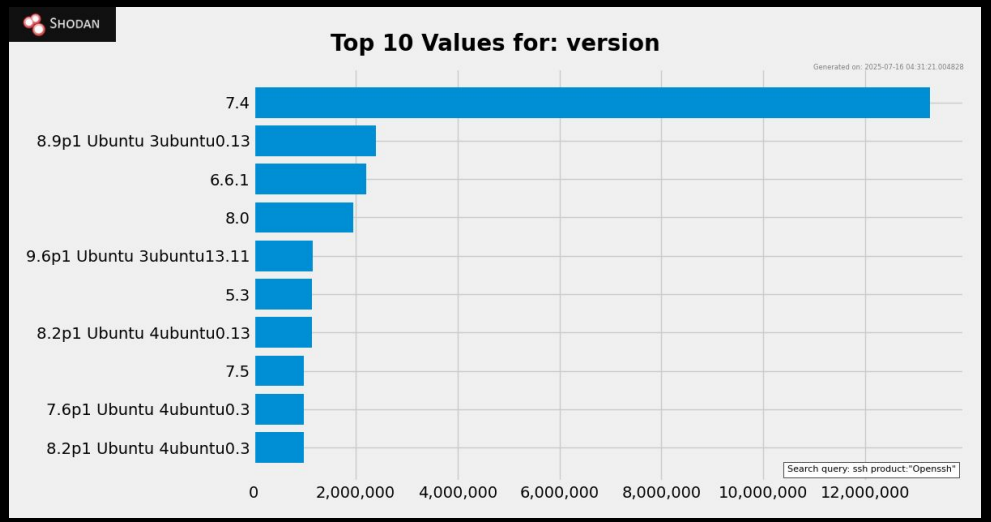
→ Exploitation of future vulnerabilities is more difficult

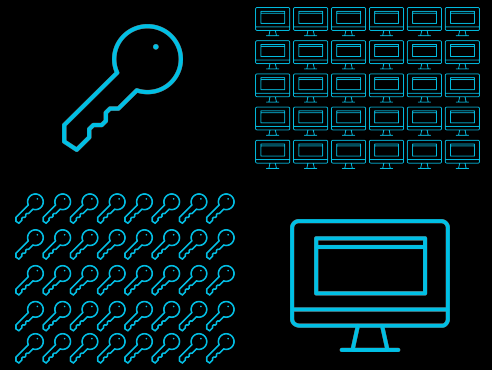
→ Slows down all sorts of automated SSH testing

→ Low adoption for newer versions

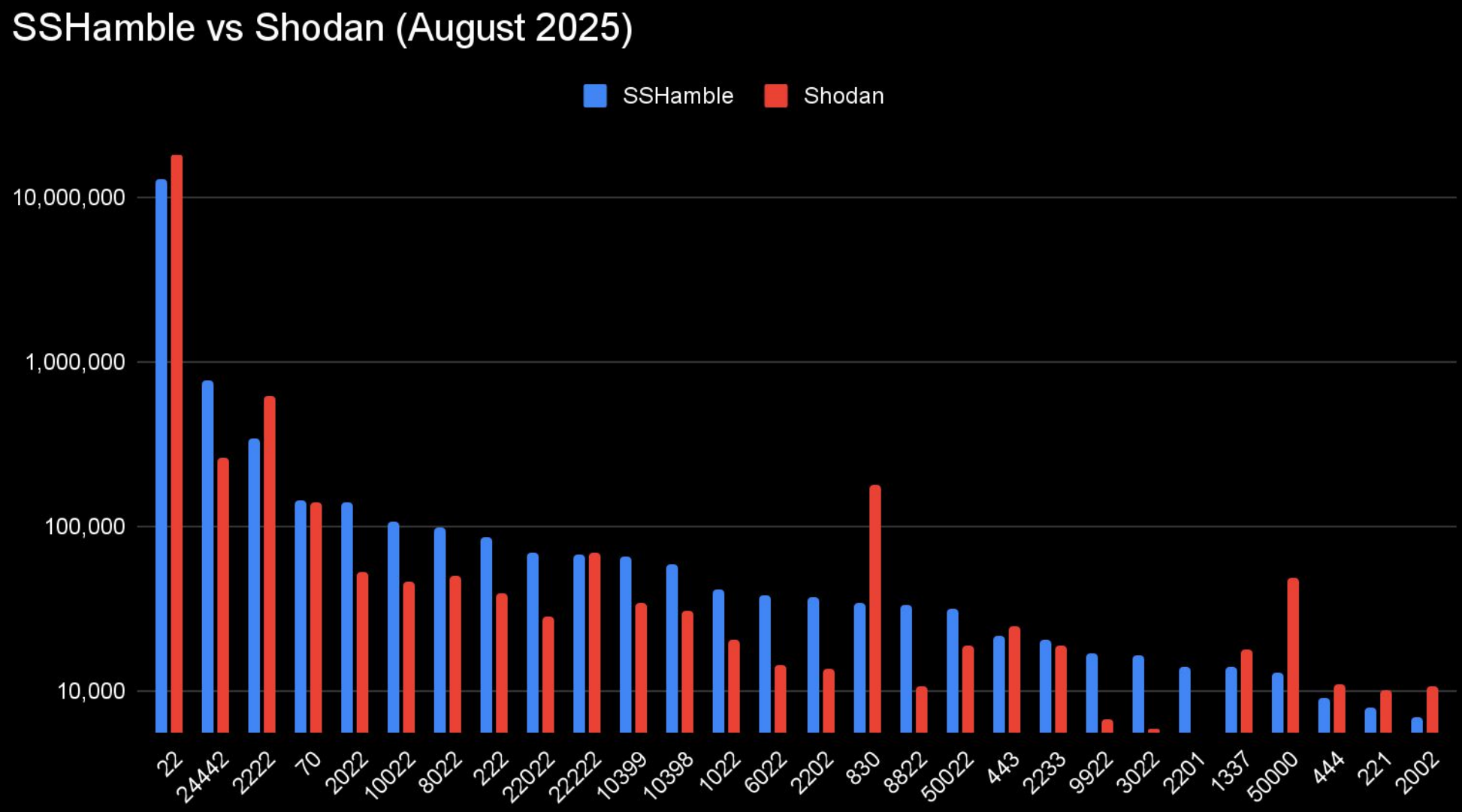
Of ~20m exposed OpenSSH servers, less than 500k are running 9.8 or newer. Stats are higher on corporate networks, but modern OpenSSH adoption is a long road.

Dropbear doesn't have anything similar and still supports high-speed tests (10k/sec/conn for pubkeys).



# IPv4 SSH ports (SSHamble vs SHODAN)



# Changes in SSH exposure (August 2024 vs 2025)

## Comparison using just port 22

→ More valid SSH servers, fewer tarpits

→ ~14.2 million reach auth state

→ ~107k resulted in a session

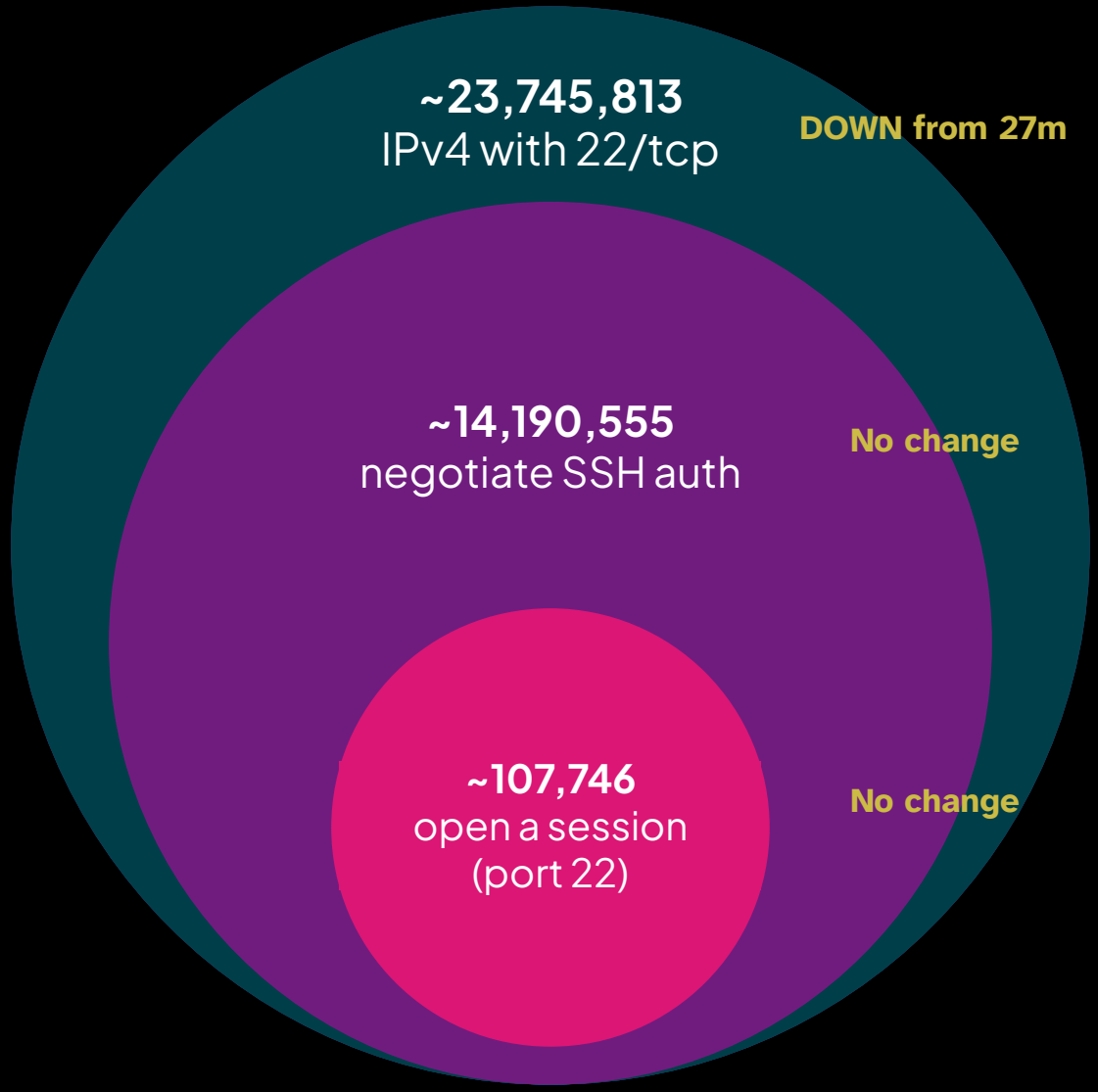
## After introducing additional ports

→ Expanded to top ~110 SSH ports

→ ~16.3 million reach auth state

→ ~20k more shells

→ New bugs!



# Little improvement overall

## Advisories and publication didn't dent exposure

→ Even more vulnerable Digi routers with auth bypass

→ Still thousands of unpatched Ruckus APs

→ Dropbear still allows unlimited pubkeys

→ Even more no-auth shells on odd ports

## Open sessions (~130k) vs real shells (~50k)

→ ~10k are obviously medium-interaction systems

→ ~17k are SonicWall firewalls with secondary auth

→ ~14k are new vulns in carrier ethernet switches

→ ~5k are quasi-sessions (limited features)

# New bugs pending disclosure (2025)

| Product | Impact |
| --- | --- |
| <Carrier Switch> | Unauthenticated shell & NETCONF via auth-method == “\x00” |
| <PBX> | Post-SSH failed login drops to an open ssh/telnet client shell |
| <Cloud Bastion Host> | ISP management shell via pubkey-any (contractually mandated) |

# Bonus vulnerabilities

**Free creds with Responder & Flamingo**

- Listen on multiple protocols and try to negotiate authentication with inbound clients
- Recommend using Responder first and then running Flamingo on the remaining ports (automatic)
- Why do this? Free credentials and early warning of investigation by your targets
- A background tcpdump can't hurt

```
$ ./Responder.py

SMB Administrator::BIDCON:...

SMB watchguard_sso::BANKOFNNN:...

SMB WGAdmin::BIGMFG:a412…

SMB _SSOWatchguard::GNRTRANSP:…

SMB PA_Agent::MYAIRNATIONAL:...
```

https://github.com/atredispartners/flamingo/

# New features in SSHamble!

- → Automatic badkeys.info blocklist lookups
- → Additional authentication bypass methods
- → Wider algorithm and host key support
- → Experimental blind exec vuln checks
- → Target filtering with --skip-versions
- → Updated go x/crypto & crypto/ forks

**https://SSHamble.com**


SSHamble v3 == v0.3.x

# BadKeys.info

## Hanno Böck's amazing key analyzer & database

→ Includes a scanner for common protocols (SSH, TLS, etc)

→ Dynamic analysis for cryptographic issues

→ Massive lookup database for known keys

→ Includes some sensitive/leaked key sets

→ Fast lookups via binary search

**https://BadKeys.Info**

# Built-in checks

| | | | |
|---|---|---|---|
| **bypass** | auth-none | skip-auth | auth-success |
| | method-null | method-empty | skip-pubkey-any |
| **publickey** | pubkey-any | pubkey-any-half | user-key |
| | half-auth-limit | pubkey-hunt | — |
| **password** | pass-any | pass-empty | pass-null |
| | pass-user | pass-change-empty | pass-change-null |
| **keyboard** | kbd-any | kbd-empty | kbd-null |
| | kbd-user | — | — |
| **gss-api** | gss-any | — | — |
| **userenum** | timing-none | timing-pass | timing-pubkey |
| **vulns** | vuln-tcp-forward | vuln-generic-env | vuln-softserve-env |
| | vuln-gogs-env | vuln-ruckus-password-escape | vuln-exec-skip-auth |
| | badkeys-blocklist | — | — |

# Getting started

```
Start a network scan
$ sshamble scan -o results.json 192.168.0.0/24

Analyze the results
$ sshamble analyze -o output results.json

Specify ports, usernames, passwords, public keys, private keys, and more
$ sshamble scan -o results.json 192.168.0.0/24 \
    --users root,admin,4DGift,jenkins \
    --password-file copilot.txt \
    -p 22,2222 \
    --pubkey-hunt-file admin-keys.pub \

Open an interactive shell for sessions
$ sshamble scan -o results.json 192.168.0.0/24 \
    --interact first --interact-auto “pty,env LD_DEBUG=all,shell”
```

# The interactive shell

```
Enter the sshamble shell with `^E`. Commands:

    exit                        - Exit the session (aliases 'quit' or '.')
    help                        - Show this help text (alias '?')
    env       a=1 b=2           - Set the specified environment variables (-w for wait mode)
    pty                         - Request a pty on the remote session (-w for wait mode)
    shell                       - Request the default shell on the session
    exec      cmd arg1 arg2     - Request non-interactive command on the session
    signal    sig1 sig2         - Send one or more signals to the subprocess
    tcp       host port         - Make a test connection to a TCP host & port
    unix      path              - Make a test connection to a Unix stream socket
    break     milliseconds      - Send a 'break' request to the service
    req       cmd arg1 arg2     - Send a custom SSH request to the service
    sub       subsystem         - Request a specific subsystem
    send      string            - Send string to the session
    sendb     string            - Send string to the session one byte at a time

sshamble>
```

# Don’t want to use a new tool?

→ We’re porting SSHamble features to Nuclei

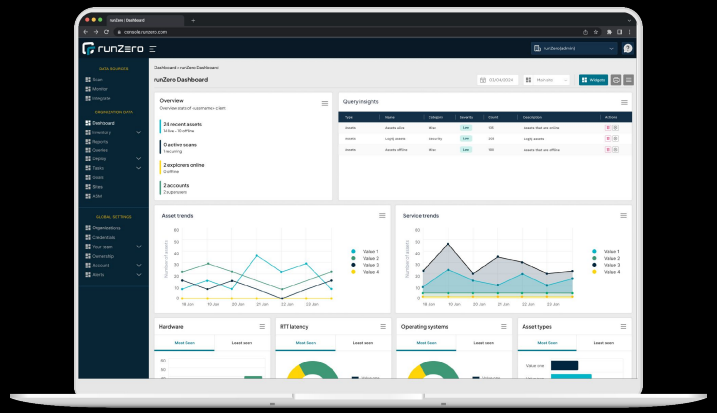
→ Soon, new SSH templates!

```
$ nuclei -v -itags ssh -target 192.168.40.254:22

                     __     _
   ____  __  _______/ /__  (_)
  / __ \/ / / / ___/ / _ \/ /
 / / / / /_/ / /__/ /  __/ /
/_/ /_/\__,_/\___/_/\___/_/   v3.4.7

                projectdiscovery.io
```

**https://github.com/projectdiscovery/nuclei**

# Thank you!

runZero.com research@runZero.com SSHamble.com




runZero RESEARCH

SSHAMBLE

# References 1/2


→ https://boehs.org/node/everything-i-know-about-the-xz-backdoor

→ https://github.com/ssh-mitm/ssh-mitm

→ https://ssh-comparison.quendi.de/comparison/hostkey.html

→ https://words.filippo.io/ssh-whoami-filippo-io/

→ https://github.com/badkeys/badkeys

→ Metasploit: ssh_identify_pubkeys (2012)

→ regreSSHion: https://www.qualys.com/2024/07/01/cve-2024-6387/regresshion.txt

→ Terrapin: https://terrapin-attack.com/

→ https://labs.watchtowr.com/auth-bypass-in-un-limited-scenarios-progress-moveit-transfer-cve-2024-5806/

→ http://thetarpit.org/2018/shithub-2018-06

→ https://helda.helsinki.fi/server/api/core/bitstreams/471f0ffe-2626-4d12-8725-2147232d849f/content

→ https://github.blog/2023-03-23-we-updated-our-rsa-ssh-host-key/

→ https://www.securityweek.com/user-id-misconfiguration-can-expose-credentials-palo-alto-networks/

# References 2/2


→ Kannisto, J., Harju, J. (2017). The Time Will Tell on You: Exploring Information Leaks in SSH Public Key Authentication. In: Yan, Z., Molva, R., Mazurczyk, W., Kantola, R. (eds) Network and System Security. NSS 2017. Lecture Notes in Computer Science(), vol 10394. Springer, Cham. https://doi.org/10.1007/978-3-319-64701-2_22

→ West, J.C., Moore, T. (2022). Longitudinal Study of Internet-Facing OpenSSH Update Patterns. In: Hohlfeld, O., Moura, G., Pelsser, C. (eds) Passive and Active Measurement. PAM 2022. Lecture Notes in Computer Science, vol 13210. Springer, Cham. https://doi.org/10.1007/978-3-030-98785-5_30

→ Neef, S. (2022). Source & result datasets for "Oh SSH-it, what's my fingerprint? A Large-Scale Analysis of SSH Host Key Fingerprint Verification Records in the DNS" [Data set]. Zenodo. https://doi.org/10.5281/zenodo.6993096

→ https://www.openwall.com/lists/oss-security/2025/04/16/2

→ https://platform.sh/blog/uncovered-and-patched-golang-vunerability/

→ https://blog.qualys.com/vulnerabilities-threat-research/2025/02/18/qualys-tru-discovers-two-vulnerabilities-in-openssh-cve-2025-26465-cve-2025-26466

→ https://badkeys.info/ & https://github.com/badkeys/badkeys

→ https://github.com/runZeroInc/sshamble

→ https://github.com/runZeroInc/excrypto

→ https://sec.cloudapps.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-cucm-ssh-m4UBdpE7

→ https://github.com/atredispartners/flamingo/