



RUNZERO

# RESEARCH


DEFCON 32

## SSHamble: Unexpected Exposures in SSH

HD MOORE | ROB KING | AUGUST 9, 2024

# Agenda

- Going Hunting
- Shaking Out Shells
- Signal Injection
- Fun with Forwarding
- Shell Injection
- Environment Control
- Broken States
- OpenSSH Fragmentation
- SSHamble



45:00

# 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**





# Going Hunting



# SSH vs SSH

→ Jia Tan targeted SSH with the XZ Utils backdoor



→ Let's target Jia Tan using SSH

# SSH public key authentication is two-stage

An SSH client can confirm if a public key is valid for a given user

→ Metasploit support since 2012, but still not widely known

```
/* XXX fake reply & always send PK_OK ? */  
/*  
* XXX this allows testing whether a user is allowed  
* to login: if you happen to have a valid pubkey this  
* message is sent. the message is NEVER sent at all  
* if a user is not allowed to login. is this an  
* issue? -markus  
*/
```

OpenSSH Source (9.8p1)

# Creating SSHamble

**A custom SSH scanner that is flexible, fast, and fun!**

- Built in Go using a mangled version of x/crypto/ssh
- Started as a half-auth public key scanner
- Evolved into an SSH research tool

```
% sshamble scan --checks pubkey-hunt \  
--pubkey-hunt-file jia.keys \  
--input-targets ipv4.txt
```

HELLO MY NAME IS

Jia Tan

I <3 Open Source!

```
$ curl https://github.com/JiaT75.keys
```

```
ssh-rsa
```

```
AAAAB3NzaC1yc2EAAAADAQABAAQCAQDHVp3Bvg/ALC61dsGehbvoqic49D4SfoiiPURSEec3/phZdAfr1hD6QSNTHLY3QDT  
b0994ZwOFi05YpUM6/qwBUAbroS64/Mp55qDBlark5v83LcTq7a29VUH3Xvu7sAgdYda16a2KnmU5lhETvBfxuS+tpGin9r  
aSp+B+z0PIpr9EmEeQgKtgKRQBiMWMtw7jBxm5INk54SmePNDva3f4m108/Z4JM76dJ7DBQGrLUqZGsRFOZclMb3YOE7DjP  
GQQ37TzGvKwLaGvRuocA8oW5zp07+uQldP2LIbt0V99eyXrgD7Wlc/sdzWeefoNltcgcV/KEg9ivD02qWFDBzAKMcJuLMhq  
xXI064KZuVjWRrflgKck5wZt0XPZ30MFqbBvjhn8zG7bIQJORMn/j6QSyHewu4Rre7uGxAuzee2PPSaSQ51dKgbdn3B3Uuw  
N8KeIO54W1VYWip+G1G2tXHZAdJOGPPaM72OAqFQBta2MzcHi3/m2HgUNBttYhSUtaeX8myfiRcnC7AphZMOuU9rrHdti2K  
D6IVArtBiorZbs8iFlzUPmdYVdeFP7EtW6EWgZSLV7rN2r2+CNVJeTrX9zA+mnRjhjq4ffgRUoQiky876kY+1YiEERm7LRB  
MkKIzM4ZsBk7VQwImSGReyfwEht9tedU5mf5pkrbL8VSMrqQQ==
```

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIFiXcmAAjTBp5kM2AUTJdAEB7DHyYuY8am8FIMROD3FG
```

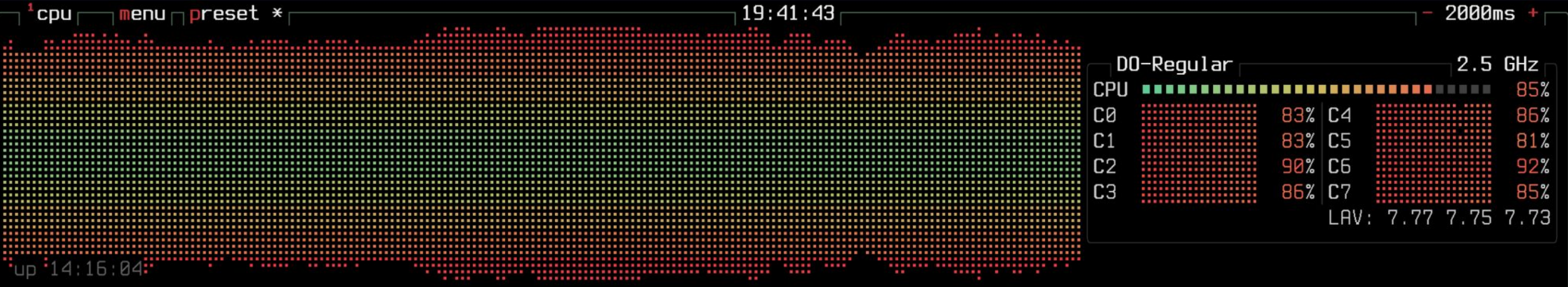


# Hunting for Jia Tan across the internet

## Putting it all together

- Copy Jia Tan's SSH public keys from GitHub
- Scan all IPv4 addresses for port 22 with zmap
- Use SSHamble to scan for key acceptance

**We got results!**



2 **mem** disks io

mem	disks	io
Total: 15.6 GiB	root 314 GiB	
Used: 1.25 GiB	10%	
Available: 14.3 GiB	Used: 14% 43.8 GiB	
Cached: 14.1 GiB	Free: 86% 270 GiB	
Free: 281 MiB	efi 123 MiB	
	10%	
	Used: 9% 11.5 MiB	
	Free: 91% 112 MiB	

4 **proc** filter per-core reverse tree < cpu lazy >

Pid	Program	Command	User	MemB	Cpu%
2815	sshamble.bin	./sshamble.bin scan - root	root	601M	85.1
46	ksoftirqd/5		root	0B	0.0
15849	bttop	bttop	root	5.8M	0.0
15	rcu_preempt		root	0B	0.0
86	kworker/5:1-mm_p		root	0B	0.0
51	ksoftirqd/7		root	0B	0.0
1238	do-agent	/opt/digitalocean/bin do-a+	root	21M	0.0
2782	sshd	sshd: root@pts/0	root	7.9M	0.0
78	kswapd0		root	0B	0.0
17705	kworker/u16:1		root	0B	0.0
17874	kworker/u16:0-ev		root	0B	0.0
1914	exim4	/usr/sbin/exim4 -bd - Debi+	root	15M	0.0
626	sshd	sshd: /usr/sbin/sshd	root	6.9M	0.0
286	systemd-journal	/lib/systemd/systemd-	root	14M	0.0
41	ksoftirqd/5		root	0B	0.0
1	systemd	/sbin/init	root	11M	0.0
613	unattended-upgr	/usr/bin/python3 /usr	root	18M	0.0
14	ksoftirqd/0		root	0B	0.0
26	ksoftirqd/2		root	0B	0.0
36	ksoftirqd/4		root	0B	0.0

↑ select ↓ info ← terminate kill signals 0/127

3 **net** sync auto zero <b eth0 n>

net	download	upload
11M	▼ 9.11 MiB/s (72.9 Mibps)	▲ 8.71 MiB/s (69.7 Mibps)
	▼ Total: 420 GiB	▲ Total: 411 GiB

HELLO MY NAME IS NOT

Jia Tan

I swear! We only scan things!

Dear Law Enforcement,

- Our scans resulted in Jia's public key hash & our IPs in everyone's logs
- Please don't arrest us!

# The ~~friends~~ shells we found along the way



## Every single result was a false positive for Jia Tan

- Tons of honeypots & broken servers
- Fixed bugs, rescanned, repeat
- 3 days later, still no Jia Tan
- Great opsec!

## Thousands of exposed systems and some fun vulnerabilities instead

- SSHamble unearthed a bunch of bugs
- Now for our actual presentation!



# SSH keys as public identities

- Public keys are used to being mostly-private
- GitHub & Launchpad changed that

```
Import SSH key
-----
Import SSH identity:  from GitHub ←
                     from Launchpad  SSH keys from GitHub or
GitHub Username:     _____
                     Enter your GitHub username.
                     [ Done      ]
                     [ Cancel   ]
```



```
ssh whoami.filippo.io
```

```
+-----+
|                                     |
|           _o/ Hello HD Moore!      |
|                                     |
| Did you know that ssh sends all   |
| your public keys to any server    |
| it tries to authenticate to?     |
|                                     |
| We matched them to the keys of   |
| your GitHub account,              |
| @hdm, which are available via the |
| GraphQL API                       |
| and at https://github.com/hdm.keys |
|                                     |
| -- Filippo (https://filippo.io) |
|                                     |
| P.S. The source of this server is |
| at https://github.com/FiloSottile/whoami.filippo.io |
|                                     |
+-----+
```

# Link a user & key to a specific server

## 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.



## Username

A list of usernames likely used by the target.

### Defaults

- root
- ec2-user
- ubuntu

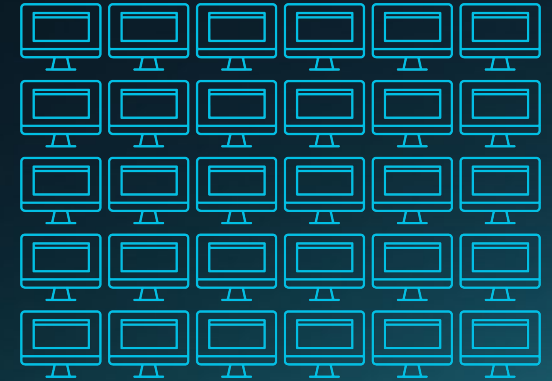
### Specific

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

# SSH public key identity primitives

## Which servers a key can access

- Test every server and every likely user for acceptance
- Fast for a small number of keys



## Which keys can access a server

- Brute force test a public key database for every likely user
- This is slow due to MaxAuthTries



## SSH servers implement MaxAuthTries

→ OpenSSH defaults to 5 & counts pubkey tests

→ This is why having >4 keys in your agent breaks

→ Not all servers count pubkey tests as failed...

# Rapid testing with a single connection

**10% of all public SSH servers do not rate limit key testing**

→ Dropbear is the most common, but many others

GlobalScape EFT	Maverick SSHD	LANCOM	Adtran
BitVise WinSSHD	GoAnywhere	Arris	Crestron
CrushFTPd	mod_sftpd	Medallia	+ Many More!



# Testing millions of public keys quickly

```
% wc -l github-2018.keys  
4,673,197 data/github.keys
```

```
% nc 192.168.68.2 22  
SSH-2.0-dropbear_2022.83
```

```
% sshamble scan --checks pubkey-hunt \ ← single connection  
--pubkey-hunt-conn-limit 1000000 --pubkey-hunt-file github-2018.keys \  
-u root 192.168.68.2  
192.168.68.2:22 pubkey-hunt is running with 4673197 test keys  
192.168.68.2:22 pubkey-hunt completed 4673190/4673197 keys in 7m37s (10544/s)  
192.168.68.2:22 pubkey-hunt accepted hunted half-auth for root with key ssh-rsa  
AAAAB3NzaC1yc2EAAAADAQABAAQADipNPRHvHknF6WLl7oEPoxxH7k13iKA/14yiWwOwHAUFg+1tl...  
  
dropbear[2921]: Exit before auth from <192.168.68.1:50311>: Exited normally
```

# Compare vs OpenSSH MaxAuthLimit=5

```
% wc -l github-2018.keys  
4,673,197 data/github.keys
```

```
% nc 192.168.68.2 2222  
SSH-2.0-OpenSSH_9.2p1 Debian-2+deb12u3
```

```
% sshamble scan --checks pubkey-hunt \ ← single connection  
--pubkey-hunt-conn-limit 1000000 --pubkey-hunt-file github-2018.keys \  
-u root 192.168.68.2 -p 2222  
192.168.68.2:2222 pubkey-hunt is running with 4673197 test keys  
192.168.68.2:2222 pubkey-hunt completed 4673190/4673197 keys in 9h50m4s (132/s)  
192.168.68.2:2222 pubkey-hunt accepted hunted half-auth for root with key ssh-rsa  
AAAAB3NzaC1yc2EAAAADAQABAAQADipNPRHvHknF6WLl7oEPoxxH7k13iKA/14yiWwOwHAUFg+1tl...  
  
sshd[6530]: Connection closed by authenticating user root 192.168.68.1 [preauth]
```

# Shaking Out Shells

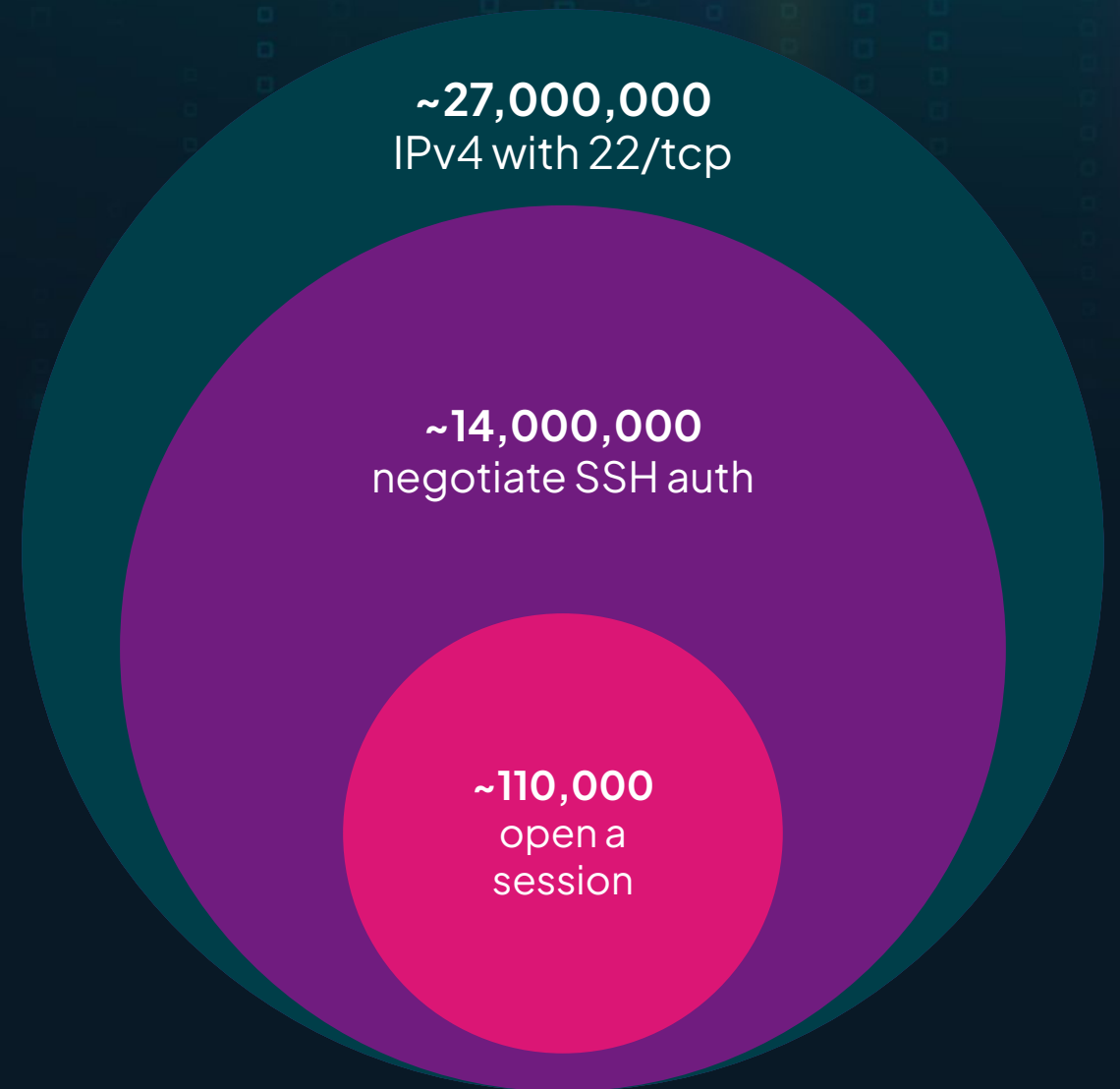


# Shaking out the shells

## A lot of broken SSH on the internet

- Tons of tarpits & buggy systems
- ~14 million reach ssh-auth state
- ~110k resulted in a session

What is all this stuff?



# Post-session authentication

Various products allow **none** authentication & then implement interactive login in the session.

Dangerous due to the extensive post-auth attack surface of SSH.

## Post-session capabilities

shell	exec
pty-req	x11-req
subsystem	env
break	signal
agent-auth-req	window-change



# Post-session authentication

```
root@          password:  
  
Copyright (c) 2021 SonicWall, Inc.  
  
Using username 'root'.  
Password: █
```

```
Please login: █
```

```
Copyright (c) 2002 - 2013 Juniper Networks, Inc. All rights reserved.
```

```
Username: █
```

# Signal Injection



# Signal handling varies by service

- OpenSSH restricts signals to relatively safe options
- Dropbear allows just about anything, even SEGV
- Signal-based attacks seem promising

Login:

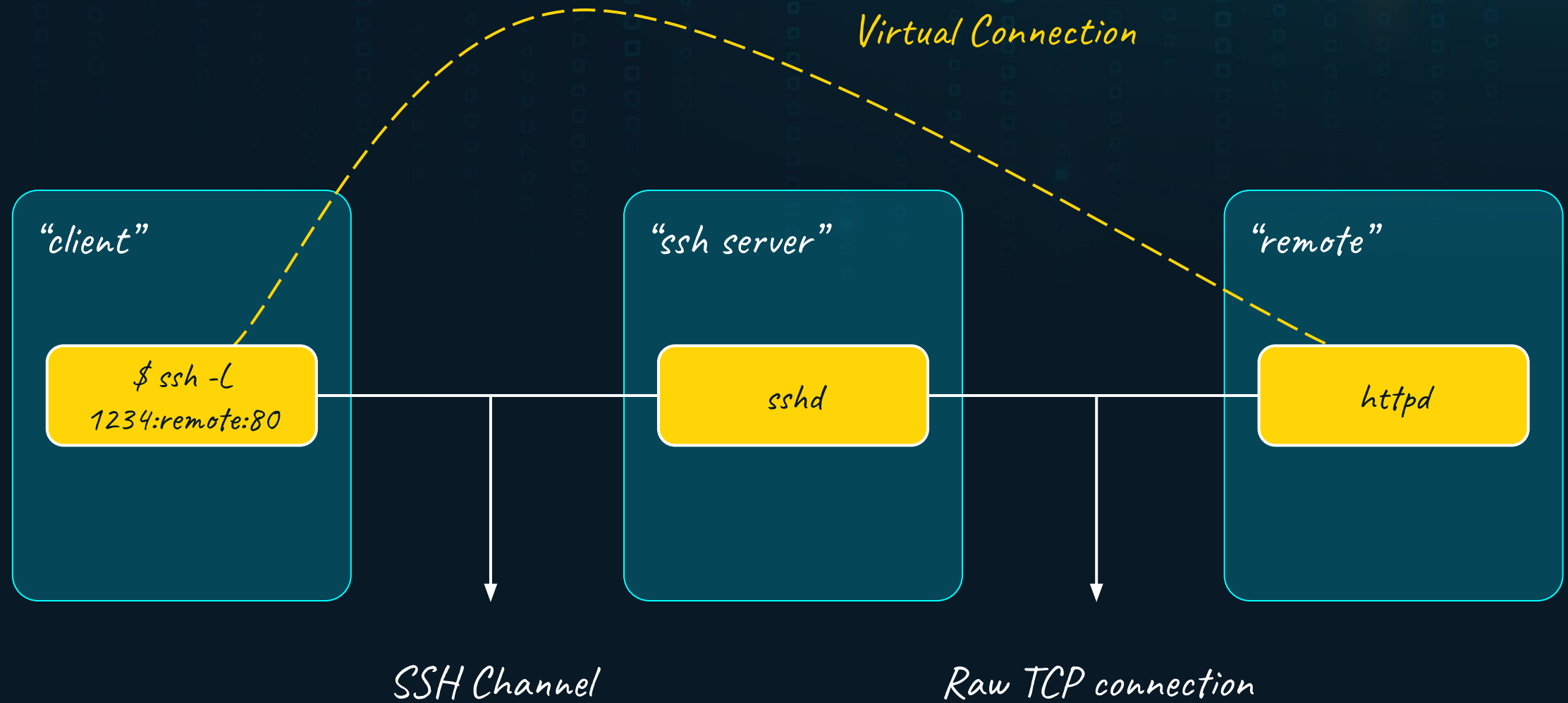
```
sshamble> signal SEGV
```

```
Aiee, segfault! You should probably report this as a bug to the developer
```

# Fun with Forwarding



# SSH connection forwarding





# Forwarding in restricted shells



## Inadvertent forwarding in SSH is a common issue

- Network devices, virtual machines, & appliances
- Can enable other attacks & bypass restrictions
- Exposes localhost-bound daemons

## Post-auth login enables unauthenticated attackers

- Not super common, but we found some anyways
- Requires testing a few destinations to evade ACLs

# ION Networks Service Access Point



# Shell Injection



# Ruckus Wireless AP command injection

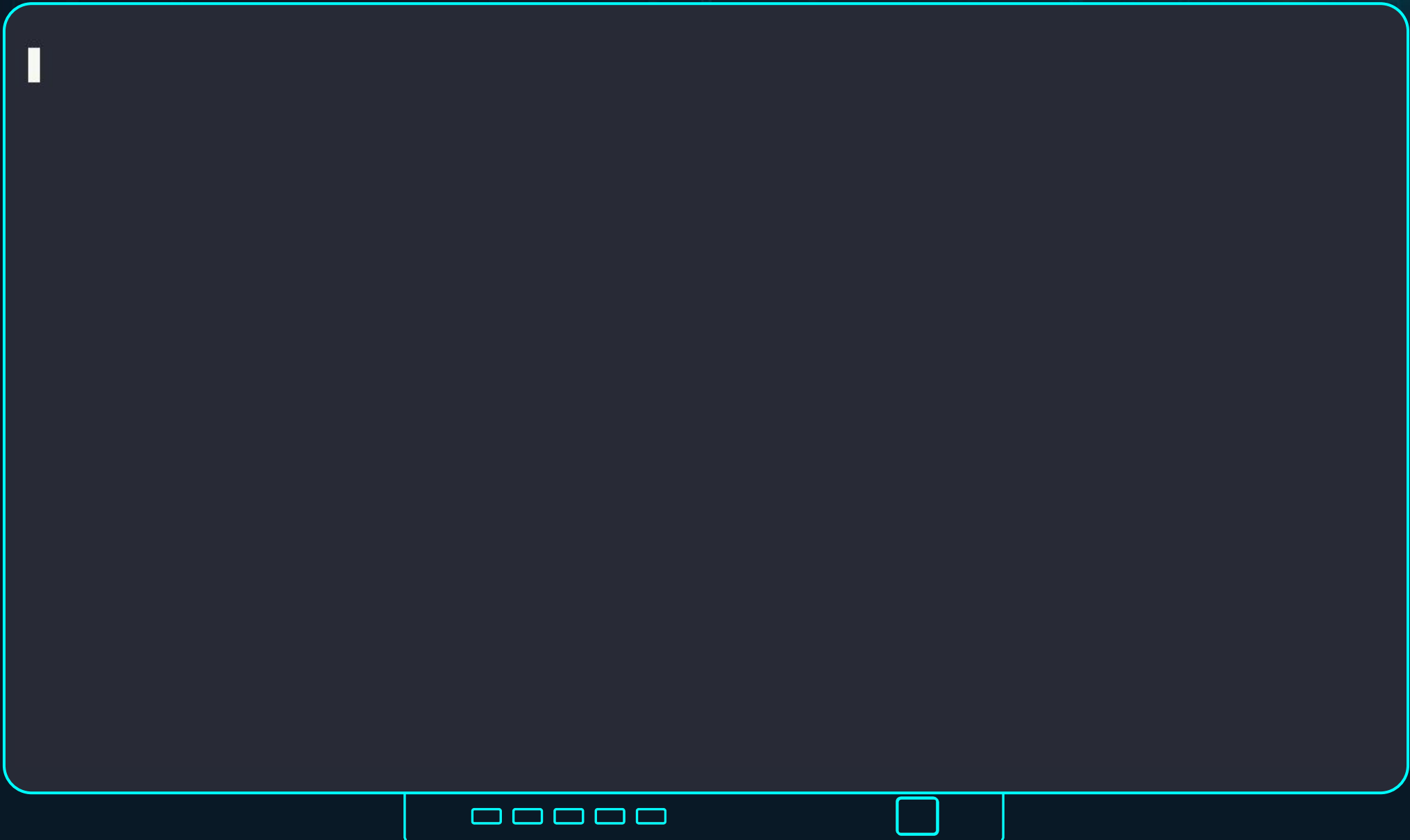
## SSH auth *none* drops to an interactive login session

- The password input is passed into a shell without escapes  
`echo -n "$(echo pa55w0rd 1>&2)" | sha256sum`

## Fixed in firmware versions v5.2.1 (stable) & 6.2.1 (tech)

- Trivial root & still ~900 exposed on the internet
- No CVE, no security mention in the release notes
- Why did this bug live so long?

# Ruckus Wireless AP command injection



# Environment Control





# Git-based code forges support SSH

- Services like GitHub, Gitlab, Bitbucket
- Projects like GOGS, Gitea, Forgejo, Gerrit
- Libraries like charmbracelet/ssh & Mina

Subject	Owner	Reviewers	Repo	Branch	Updated	Size	Status	CR	V	CS	FV
Add query limit to listProjects RestAPI with no parameters	José Granha	Dandan, Luca	gerrit	master	10:31 AM	3	3 missing	1	1		
Fix compilation and test errors after remotes' API merge	Darek	Tony, Dandan, +1	plugins/pull-replication	master	10:29 AM	2	2 missing	1			
TraceIT: Speed up noAutoRetry(ExceptionCausesNormalRetrying)	Edwin	Patrick	gerrit	master	Jul 26	1	1 missing				
Remove unnecessary usage of LazyArgs for logging	Edwin	Patrick	gerrit	master	Jul 26	1	1 missing				
Stop using LazyArgs for logging operation metadata	Edwin	Patrick	gerrit	master	Jul 26	1	1 missing				
Implement Bazel build	davido	Matthias, Saša, +2	k8s-gerrit	master	Jul 26	4	4 missing				
Drop remaining debug logs for known groups	Edwin	Patrick	gerrit	master	Jul 26	1	1 missing				
Disallow tracing configs that trigger tracing for too many requests	Edwin	Patrick	gerrit	master	Jul 26	1	1 missing				
Warn about too broad tracing configs	Edwin	Patrick	gerrit	master	Jul 26	1	1 missing				
PerformanceMetrics: Use cfg section that doesn't conflict with tra...	Edwin	Patrick	gerrit	master	Jul 26	1	1 missing				
RestApiServlet: Remove usage of LazyArgs to log response JSON	Edwin	Patrick	gerrit	master	Jul 26	1	1 missing				
[Operator] Move Constants class to API package	davido	Matthias, Saša, +2	k8s-gerrit	master	Jul 26	3	3 missing				
[Operator] Compute labels in dedicated factory	Thomas Dräbl...	Matthias, Saša, +1	k8s-gerrit	master	Jul 26	4	4 missing				
[Operator] Create components for NFS workaround in dedicated fa...	Thomas Dräbl...	Matthias, Saša, +1	k8s-gerrit	master	Jul 26	3	3 missing				
[Operator] Add missing hashCode() method to KafkaConfig	Thomas Dräbl...	Matthias, Saša, +1	k8s-gerrit	master	Jul 26	3	3 missing				
[Operator] Remove circular dependency during probe creation	davido	Matthias, Saša, +2	k8s-gerrit	master	Jul 26	4	4 missing				
[Operator] Create VolumeMounts for shared Volume in dedicated f...	Thomas Dräbl...	Matthias, Saša, +1	k8s-gerrit	master	Jul 26	4	4 missing				

forgejo-contrib / delightful-forgejo

Code Issues 6 Pull requests Activity

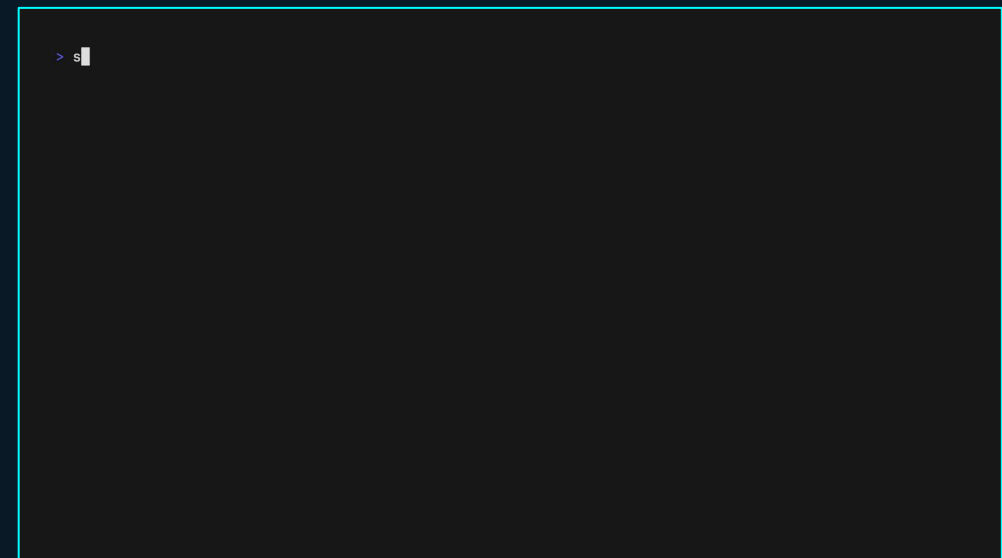
A curated list of delightful Forgejo-related projects and resources. <https://delightful.club/delightful-forgejo/>

awesome awesome-list delightful delightful-list forge forgejo git

87 commits 1 branch 0 tags 231 KIB

main Find a file HTTPS <https://codeberg.org/forgejo-contrib/delightful-forgejo.git>

- Ikuyo Kita 881faf7afa Add Kita to delightful-contributors.md 2 weeks ago
- resources use SVG for Forgejo icon last year
- .editorconfig add editorconfig last year
- delightful-contributors.md Add Kita to delightful-contributors.md 2 weeks ago
- LICENSE initialise delightful repo last year
- README.md Add Codejikka 2 weeks ago



# Gitlab, Gitea, & Forgejo

- Environment control limited to **GIT\_PROTOCOL**
- Git only parses the **version** parameter
- Usually safe, but bugs still exist
  - Go < 1.19.3 via [CVE-2022-41716](#)

```
GIT_PROTOCOL=version=2:\x00PATH=C:\Users\gitlab\repositories\rob
```

# GOGS “env” command injection

## GOGS was the first Go-based git forge



- Supports SSH “env”, but gets it terribly wrong

```
ExecCmd("env", fmt.Sprintf("%s=%s", env.Name, env.Value))
```

## This does nothing, “env” doesn't set the parent env

- GOGS supports self-registration & **env** often supports **-S**
- Exploit with env `-SA=B touch /tmp/fun`
- No patch available, consider alternatives

\* Independently discovered by Sonar Source (reported 2 days before us): CVE-2024-39930

# SSH libraries & env: Apache Mina

## Apache Mina is a Java package for SSH clients & servers

- Passes "env" variables to caller with no restrictions
- Callers (like Gerrit) **do** limit the environment
- JGit & friends don't spawn subprocesses

```
✓ J AbstractGitCommand.java java/com/google/gerrit/sshd 1
String gitProtocol = env.getEnv().get(GIT_PROTOCOL);

✓ J ShowCaches.java java/com/google/gerrit/sshd/commands 1
String s = env.getEnv().get(Environment.ENV_COLUMNS);

✓ J ShowConnections.java java/com/google/gerrit/sshd/commands 1
String s = env.getEnv().get(Environment.ENV_COLUMNS);

✓ J ShowQueue.java java/com/google/gerrit/sshd/commands 1
String s = env.getEnv().get(Environment.ENV_COLUMNS);
```

# SSH libraries & env: Soft Serve

**Soft Serve is a feature–full Git forge written in Go**

- Uses charmbracelet/ssh (a gliderlabs/ssh fork)
- Accepts all environment variables
- Soft Serve passes these to Git
- Combination is a remote shell

**CVE-2024-41956**



# Remote Code Execution in Soft Serve





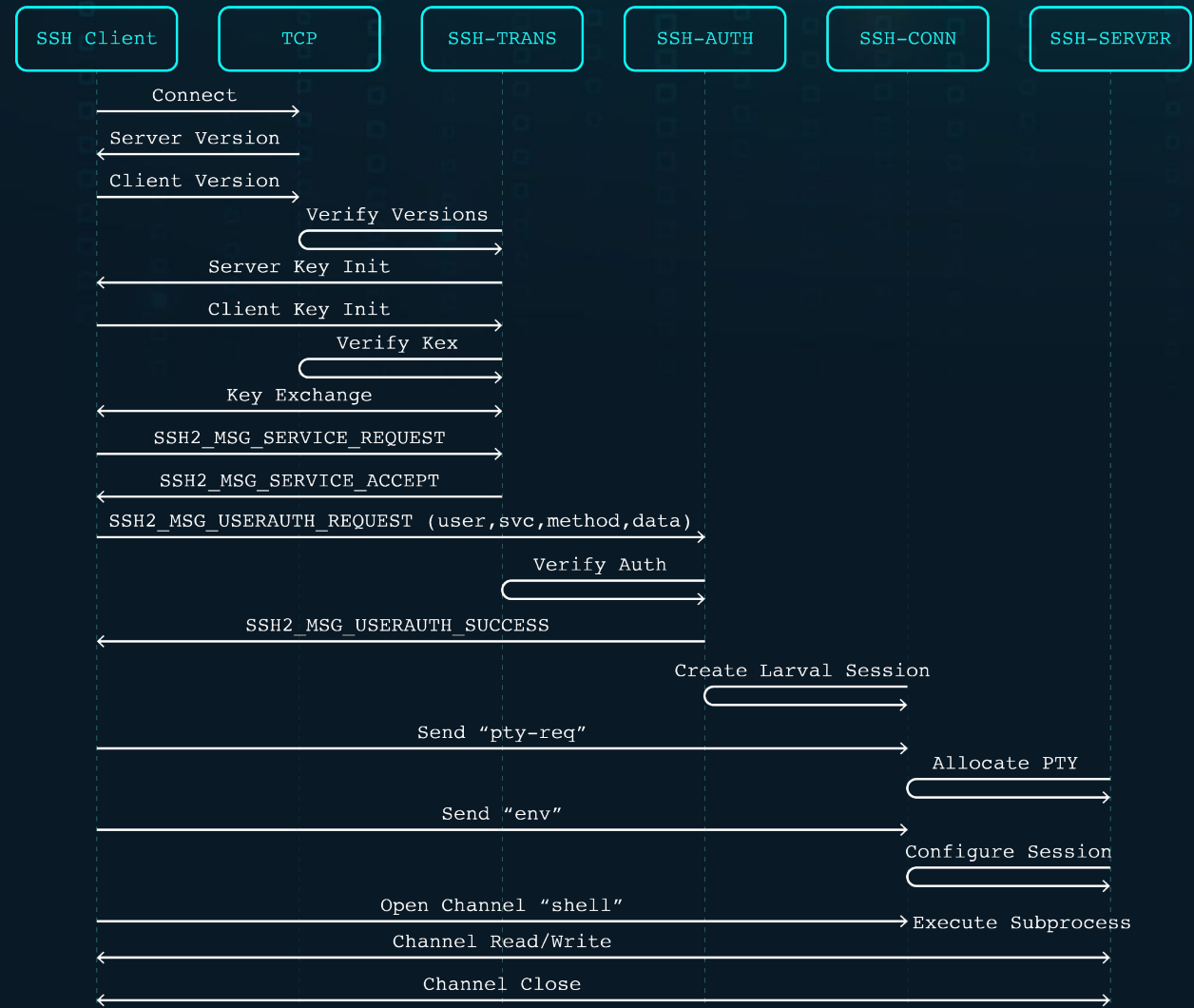
# Broken States



# Secure shell uses a strict state engine

- Accepted client message types change as the connection moves through each state
- OpenSSH & Dropbear remap the table of command handlers on each state change
- Message IDs are clamped to specific allowed ranges by session state

SSH2_MSG_TRANSPORT_MIN	1	
SSH2_MSG_TRANSPORT_MAX	49	
SSH2_MSG_USERAUTH_MIN	0	
SSH2_MSG_USERAUTH_MAX	79	
SSH2_MSG_USERAUTH_PER_METHOD_MIN		60
SSH2_MSG_USERAUTH_PER_METHOD_MAX		79
SSH2_MSG_CONNECTION_MIN	80	
SSH2_MSG_CONNECTION_MAX	127	
SSH2_MSG_RESERVED_MIN		128
SSH2_MSG_RESERVED_MAX		191
SSH2_MSG_LOCAL_MIN	192	
SSH2_MSG_LOCAL_MAX	255	
SSH2_MSG_MIN	1	
SSH2_MSG_MAX	255	

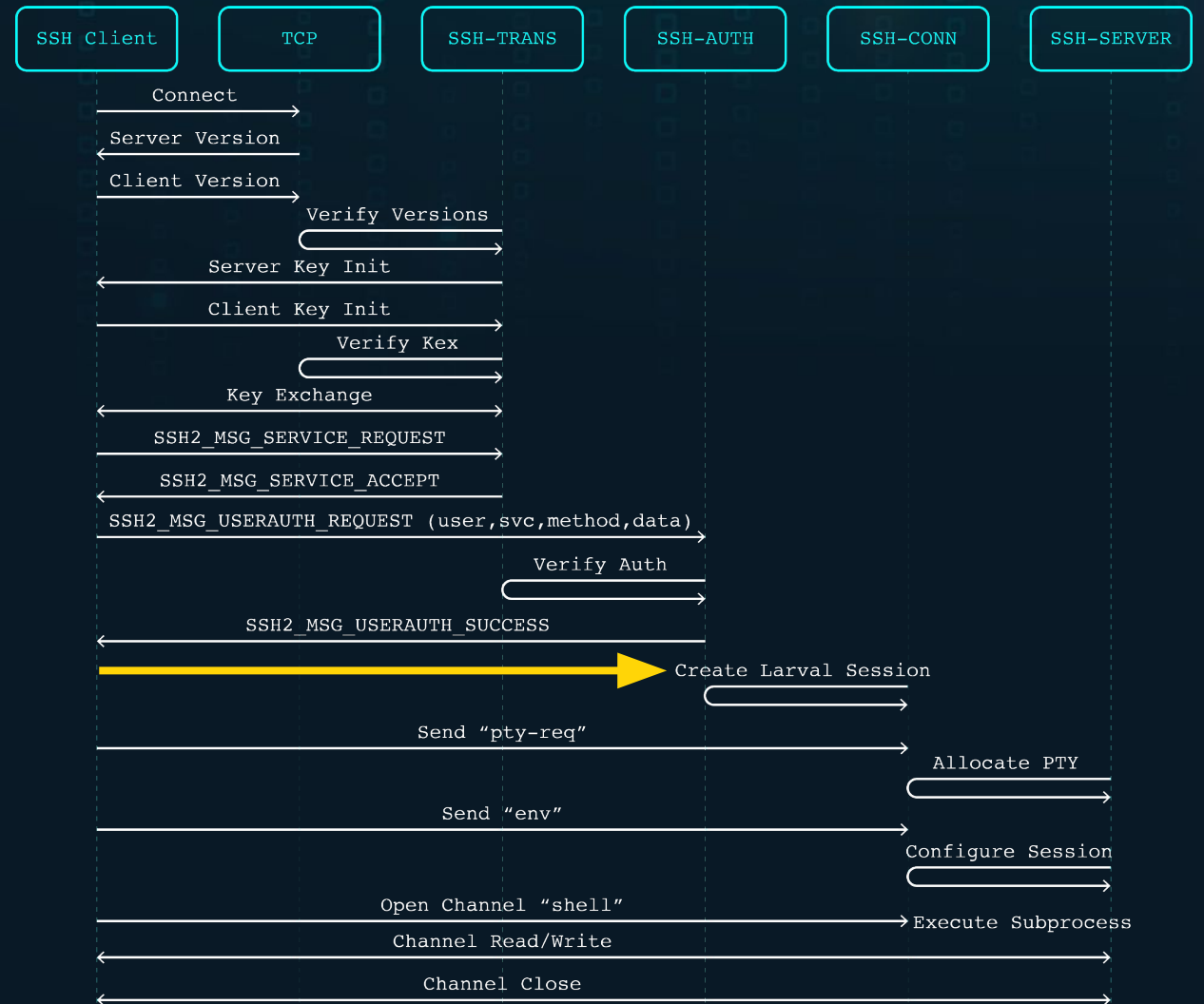


# State transitions gone wrong (historic)

## CVE-2018-10933

A bug in libssh where the server trusted a client-sent USERAUTH\_SUCCESS message.

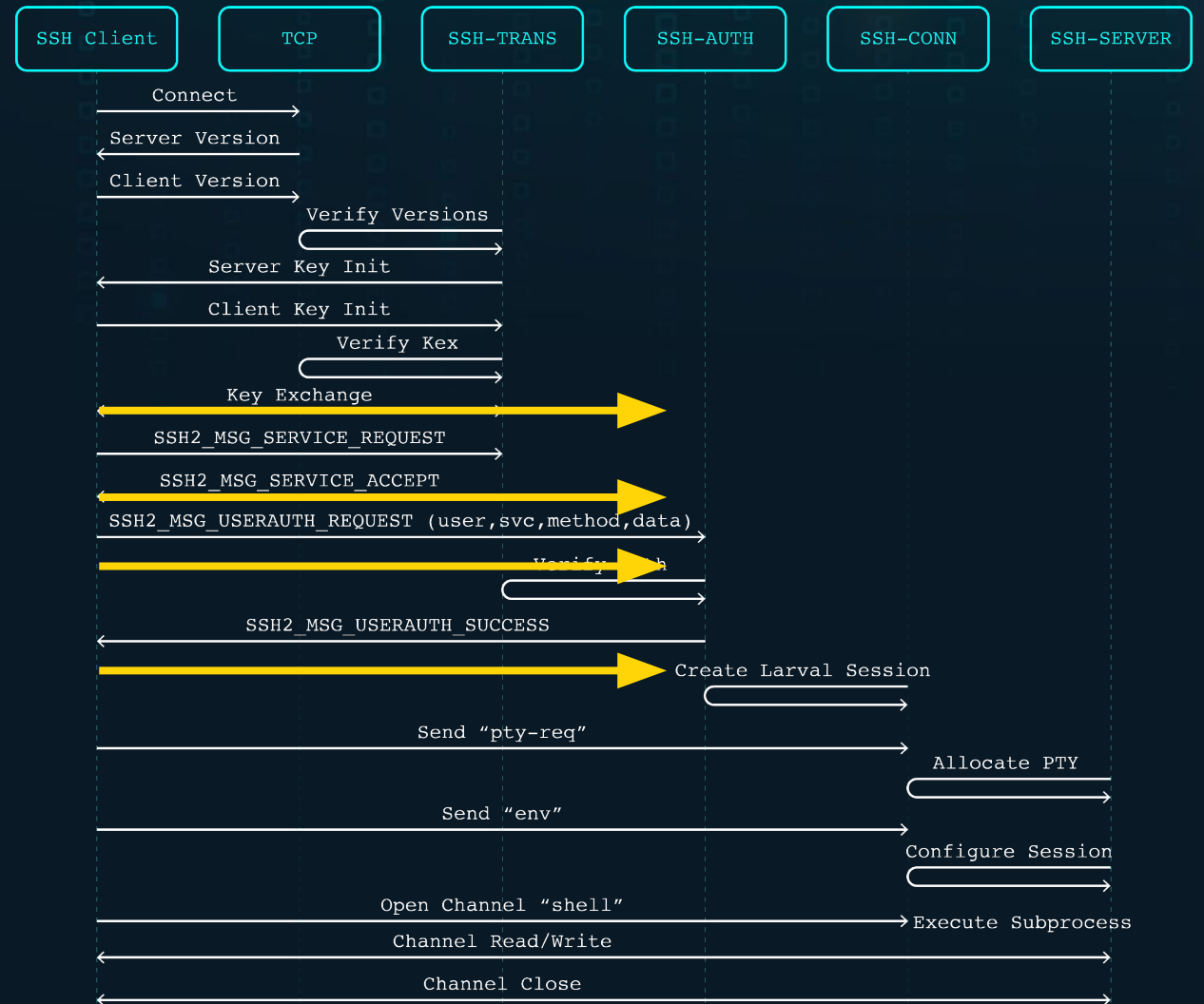
Metasploit support!



# State transitions gone wrong (new)

What happens if we ask for a session at every possible state transition?

Free shells!



# State transition vulnerabilities

Product	Impact	Details
Digi TransPort WR Gateways	Remote CLI as SUPER	Authentication bypass due to uninitialized variable. Updates available for WR11, WR21, WR31, WR44R, WR44RR included in version 8.6.0.4. The Digi International product security team was great to work with (via Bugcrowd).
Realtek ADSL Routers	Remote CLI access as admin	Authentication bypass via skipping ssh-userauth. White-labeled by Netis, Neterbit, and many other vendors. Observed in firmware as recent as 2023.
Panasonic Ethernet Switches	Remote CLI access as admin	Authentication bypass via skipping auth "none" after the ssh-userauth sequence. Models include PN28080K, PN28240i, and likely others.

# Neterbit NSL-224 authentication bypass








# Digi TransPort authentication bypass



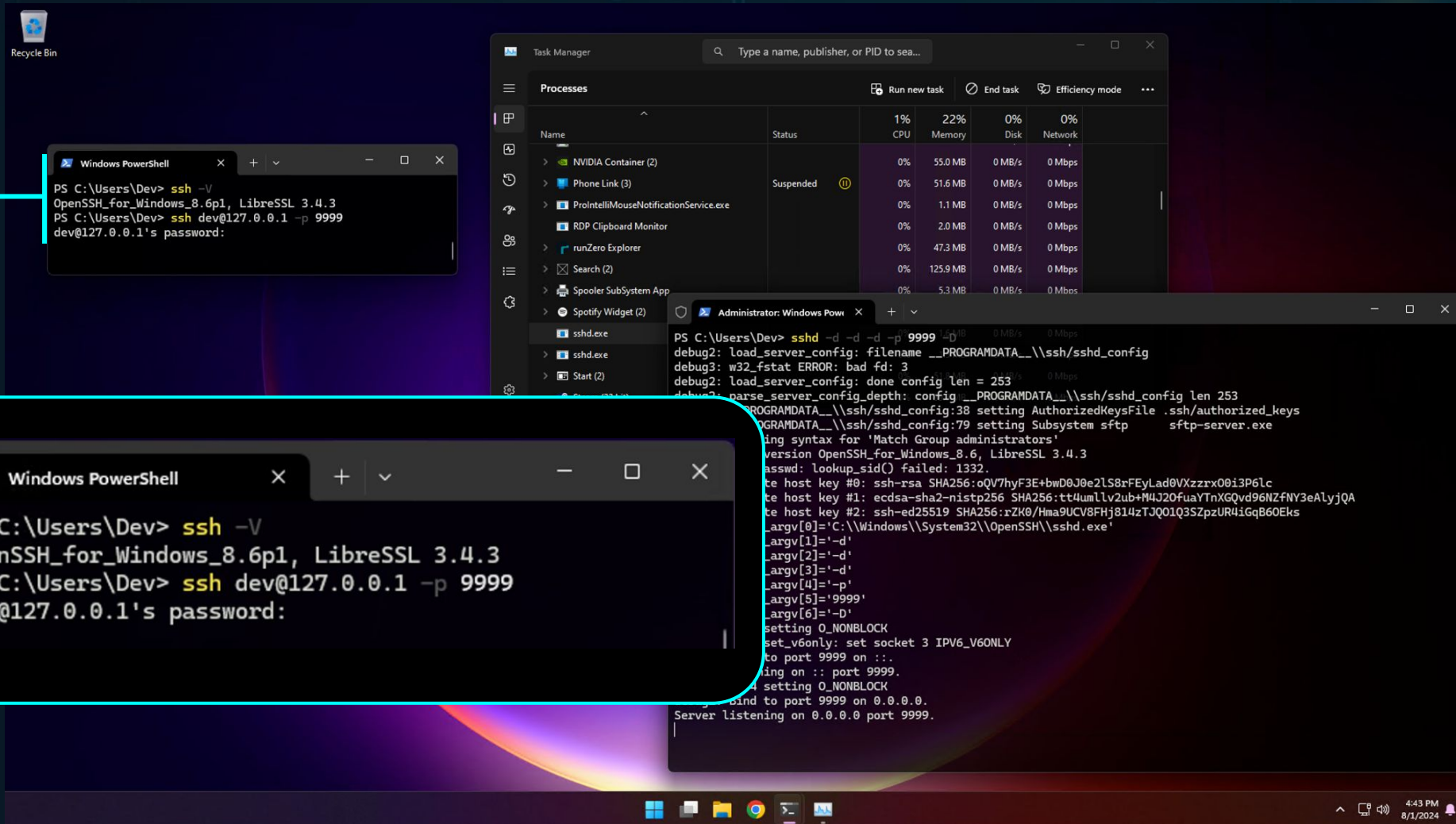
# OpenSSH Fragmentation



# OpenSSH divergence by platform

Name	Divergence	Notes
Apple macOS	Light	Changes are limited to macOS compatibility, support for the Keychain, the macOS PKCS helper, & endpoint event logging support.
Debian/Ubuntu Linux	Moderate	Systemd support & much more (36+ patches) 
Red Hat Linux	Moderate	Systemd support & much more (~60 patches) 
PKI-X SSH	Major	Forked in 2002 for X509 support, commonly found in networking gear and FIPS-compliant network appliances. Generally follows OpenSSH changes, but not exactly.
Microsoft Windows	Extreme	Over 350 files changed. Replaces fork with subprocesses, removes chroot support & log sanitization. Logs to Windows Events. Sends telemetry containing SSH-encrypted values. Password authentication uses Lsa* functions. Still hasn't fixed Terrapin. Not affected by regreSSHion. 

# OpenSSH for Windows



The screenshot illustrates the OpenSSH for Windows environment. It features a Task Manager window showing the 'Processes' tab with 'sshd.exe' running. Two PowerShell windows are shown: one for client usage and one for server startup logs.

**Client PowerShell Window:**

```
PS C:\Users\Dev> ssh -V
OpenSSH_for_Windows_8.6p1, LibreSSL 3.4.3
PS C:\Users\Dev> ssh dev@127.0.0.1 -p 9999
dev@127.0.0.1's password:
```

**Server PowerShell Window (Administrator):**

```
PS C:\Users\Dev> sshd -d -d -p 9999
debug2: load_server_config: filename _PROGRAMDATA_\\ssh/sshd_config
debug3: w32_fstat ERROR: bad fd: 3
debug2: load_server_config: done config len = 253
debug3: parse_server_config_depth: config _PROGRAMDATA_\\ssh/sshd_config len 253
debug3: parse_server_config: config _PROGRAMDATA_\\ssh/sshd_config:38 setting AuthorizedKeysFile .ssh/authorized_keys
debug3: parse_server_config: config _PROGRAMDATA_\\ssh/sshd_config:79 setting Subsystem sftp sftp-server.exe
debug3: parse_server_config: warning syntax for 'Match Group administrators'
debug3: parse_server_config: version OpenSSH_for_Windows_8.6, LibreSSL 3.4.3
debug3: passwd: lookup_sid() failed: 1332.
debug3: set host key #0: ssh-rsa SHA256:oQV7hyF3E+bwD0J0e2LS8rFEyLad0VXzzrx00i3P6lc
debug3: set host key #1: ecdsa-sha2-nistp256 SHA256:tt4umllv2ub+N4J20fuaYTnXGQvd96MZFNy3eAlyjQA
debug3: set host key #2: ssh-ed25519 SHA256:rZK0/Hma9UCV8FHj814zTJQ01Q3SZpzUR4iGqB60Eks
debug3: set _argv[0]='C:\\Windows\\System32\\OpenSSH\\sshd.exe'
debug3: set _argv[1]='-d'
debug3: set _argv[2]='-d'
debug3: set _argv[3]='-d'
debug3: set _argv[4]='-p'
debug3: set _argv[5]='9999'
debug3: set _argv[6]='-D'
debug3: setting 0_NONBLOCK
debug3: set_v6only: set socket 3 IPV6_V6ONLY
debug3: listening on port 9999 on ::.
debug3: listening on :: port 9999.
debug3: setting 0_NONBLOCK
debug3: bind to port 9999 on 0.0.0.0.
Server listening on 0.0.0.0 port 9999.
```

# OpenSSH for Windows Telemetry

- OpenSSH for Windows sends detailed usage data to Microsoft
- Client and server versions, kex init parameters, auth methods

```
void send_ssh_version_telemetry (const char* ssh_version,
    const char* peer_version, const char* remote_protocol_error)
{
    TraceLoggingRegister (g_hProvider1);
    TraceLoggingWrite (
        g_hProvider1,
        "Startup",
        TelemetryPrivacyDataTag (PDT_ProductAndServiceUsage),
        TraceLoggingKeyword (MICROSOFT_KEYWORD_MEASURES),
        TraceLoggingString (ssh_version, "ourVersion"),
        TraceLoggingString (remote_protocol_error, "remoteProtocolError"),
        TraceLoggingString (peer_version, "peerVersion")
    );
    TraceLoggingUnregister (g_hProvider1);
}
```

# compat/timingsafe\_bcmp.c

```
int timingsafe_bcmp(const void *b1, const void *b2, size_t n) {
    const unsigned char *p1 = b1, *p2 = b2;
    int ret = 0;
    for (; n > 0; n--) {
        ret |= *p1++ ^ *p2++;
    }
    return (ret != 0);
}
```

## A solid bit of code from DJM

- Timing-safe
- Efficient
- Secure



# compat/timingsafe\_bcmp.c for Windows

```
int timingsafe_bcmp(const void *b1, const void *b2, size_t n) {
    const unsigned char *p1 = b1, *p2 = b2;
    int ret = 0;
    for (; n > 0; n--) {
#ifdef WINDOWS
        if (*p1 == '\\r' && *(p1 + 1) == '\\n' && *p2 == '\\n')
            p1++;
#endif // WINDOWS
        ret |= *p1++ ^ *p2++;
    }
    return (ret != 0);
}
```

# compat/timingsafe\_bcmp.c for Windows

```
int timingsafe_bcmp(const void *b1, const void *b2, size_t n) {
    const unsigned char *p1 = b1, *p2 = b2;
    int ret = 0;
    for (; n > 0; n--) {
#ifdef WINDOWS
        if (*p1 == '\r' && *(p1 + 1) == '\n' && *p2 == '\n')
            p1++;
#endif // WINDOWS
        ret |= *p1++ ^ *p2++;
    }
    return (ret != 0);
}
```

## Two lines, but so many bugs!

- Not timing-safe
- 1-byte OOB per \r
- Unequal byte match

# A critical function within OpenSSH

- MAC check on every SSH packet
- RSA signature verification
- SSH certificate comparison
- X11 cookie comparison
- chachapoly\_crypt() MAC
- SSHFP DNS record checks
- SSH agent validation
- WebAuthn SK checks
- SSH keygen verification
- ..and much more!

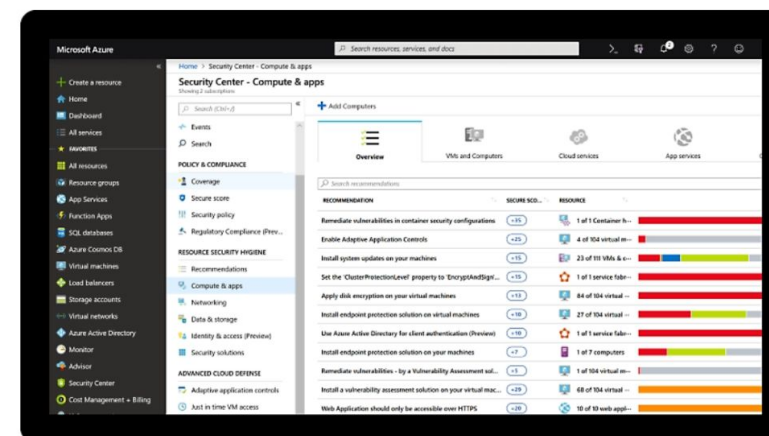
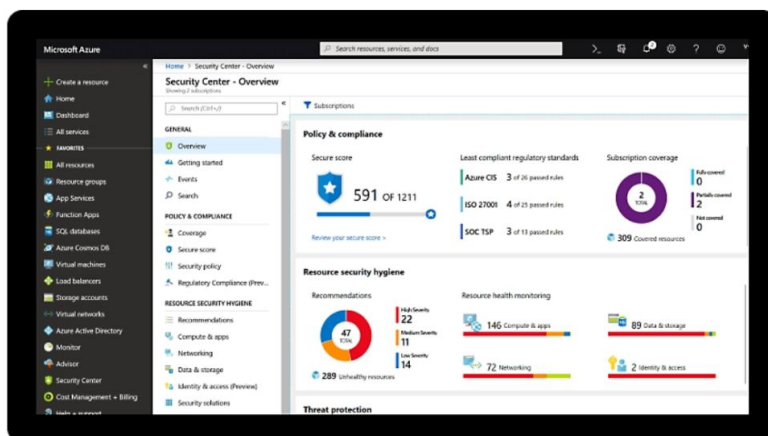
## One of the most sensitive functions, but what can we do with it?

- Attacker has limited influence on the first argument
- Requires brute force to trigger in the MAC check
- Not obviously exploitable :(

## Comprehensive security and compliance, built in

✓ Microsoft invests more than **\$1 billion annually** on cybersecurity research and development.

✓ We employ more than **3,500 security experts** who are dedicated to data security and privacy.



[Learn more about security on Azure](https://azure.microsoft.com/en-us/products/devops/server)

<https://azure.microsoft.com/en-us/products/devops/server>

# Microsoft Security Response Center



“

*Thank you again for submitting this issue to Microsoft. Although your report is valid, currently, MSRC prioritizes vulnerabilities that are assessed as “Important” or “Critical” severities for immediate servicing. After careful investigation, this case does not meet MSRC’s current bar for immediate servicing because currently it appears to be theoretical due to no control over the first argument to the function & would require a brute force style attack to obtain a single byte of data. If you can prove remote reachability or the ability to leak information remotely, then please submit a new report & we are happy to investigate this further!*

”

# SSHamble





- A research tool for SSH implementations
- Interesting attacks against authentication
- Post-session authentication attacks
- Pre-authentication state transitions
- Post-session enumeration
- Easy timing analysis

<https://SSHamble.com>



# Built-in checks

<b>bypass</b>	auth=none	skip=auth	auth=success
	method=null	method=empty	skip=pubkey-any
<b>publickey</b>	pubkey-any	pubkey-any-half	user-key
	half-auth-limit	pubkey-hunt	—
<b>password</b>	pass-any	pass-empty	pass-null
	pass-user	pass-change-empty	pass-change-null
<b>keyboard</b>	kbd-any	kbd-empty	kbd-null
	kbd-user	—	—
<b>gss-api</b>	gss-any	—	—
<b>userenum</b>	timing-none	timing-pass	timing-pubkey
<b>vulns</b>	vuln-tcp-forward	vuln-generic-env	vuln-softserve-env
	vuln-gogs-env	vuln-ruckus-password-escape	—

# 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 \  
  --privkey-hunt-file admin-keys.priv
```

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:

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

sshamble>

# Happy scanning!

|

# Vulnerabilities

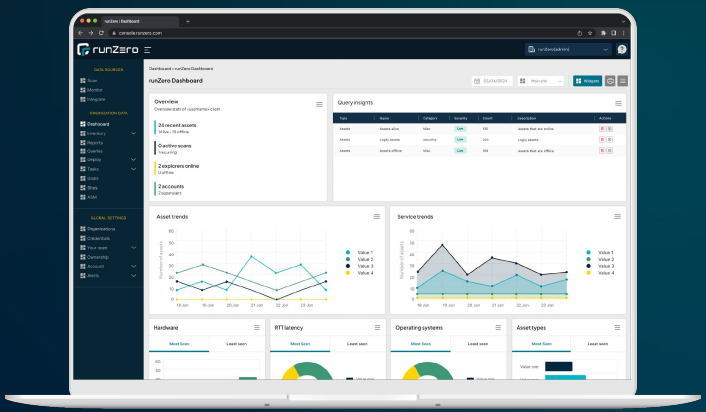
Product	Impact
Ruckus Wireless APs	Unauthenticated root command execution
Digi TransPort Gateways	Unauthenticated remote CLI access as SUPER
Panasonic Ethernet Switches	Unauthenticated remote CLI access as admin
Realtek ADSL Gateways	Unauthenticated remote CLI access as admin
Soft Serve	Authenticated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





# Thank you.

HD MOORE | ROB KING | AUGUST 9, 2024



[runZero.com](https://runZero.com)



[research@runZero.com](mailto:research@runZero.com)



[SSHamble.com](https://SSHamble.com)

# References

- <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/>
- 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](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](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>