Chhoyhopper: A Moving Target Defense with IPv6

A S M Rizvi
asmrizvi@usc.edu

John Heidemann
johnh@isi.edu

USC Viterbi School of Engineering
Information Sciences Institute

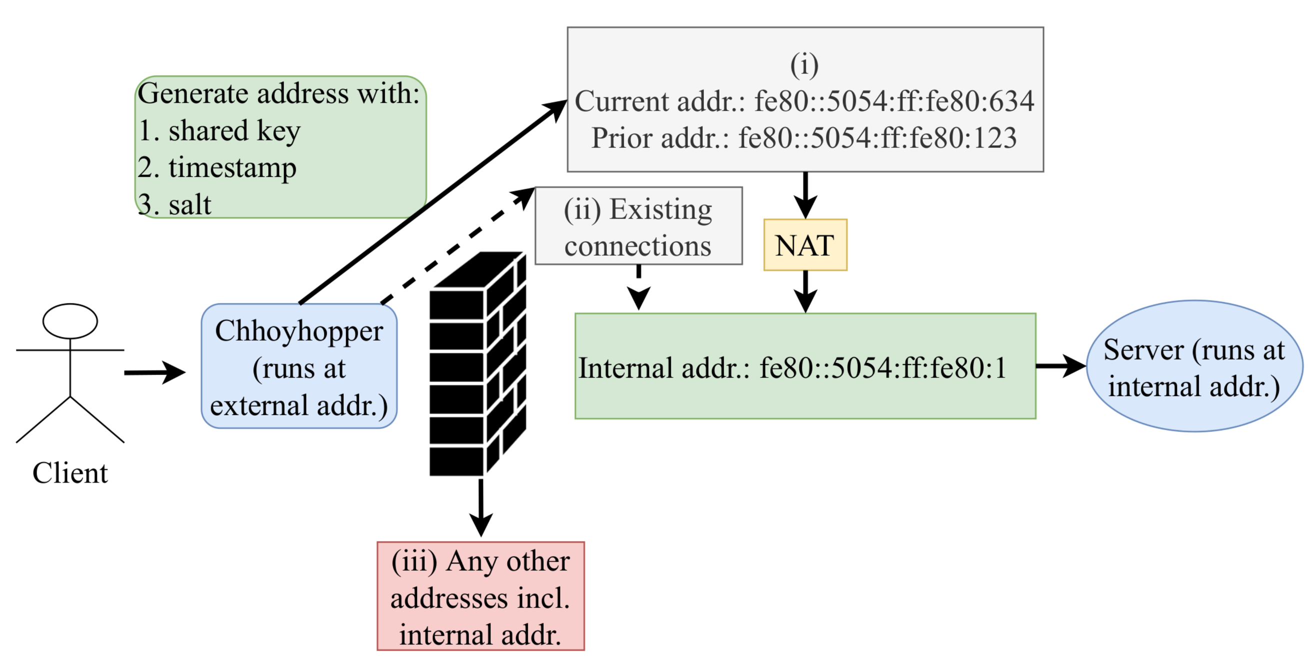
Introduction

Services on the public Internet are frequently scanned, then subject to brute-force and denial-of-service attacks. We would like to run such services stealthily, available to friends but hidden from adversaries. In this work, we propose a moving target defense named "*Chhoyhopper*" that utilizes the vast IPv6 address space to conceal publicly available services. The client and server hop to different IPv6 addresses in a pattern based on a shared, pre-distributed secret and the time of day. By hopping over a /64 prefix, services cannot be found by active scanners, and passively observed information is useless after two minutes. We demonstrate our system with the two important applications—SSH and HTTPS.

This poster presents the design and implementation of Chhoyhopper with SSH and HTTPS applications.

Chhoyhopper Design

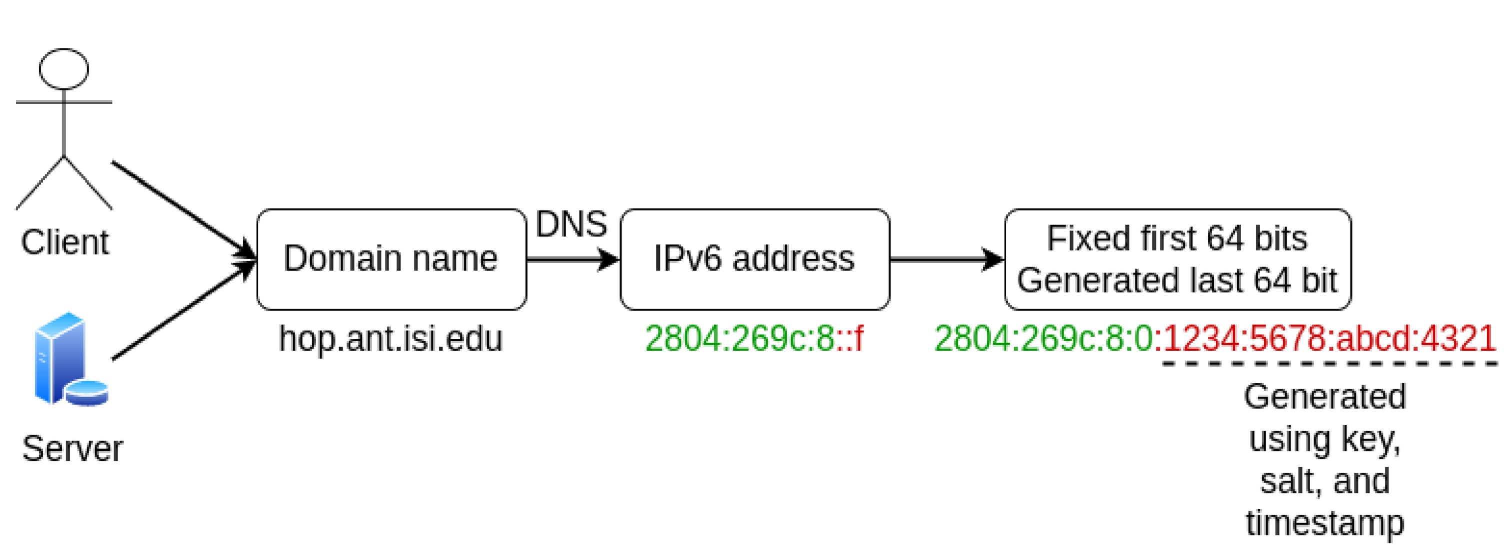
Our goal is to allow the client to rendezvous with the server on a public, but temporary IPv6 address. By allocating the temporary address from a large space (2^64 addresses), scanning is impractical.



Address hopping pattern:

- Client and server must follow the same hopping pattern to rendezvous.
- Client and server share a pre-distributed key and salt value.
- The server and the client compute the same temporary address by computing a cryptographic hash (we use SHA-256) of the shared secret, a salt value, and the current time in minutes.
- Using NAT, (i) only the current IPv6 address will be translated to the internal service address, (ii) we keep the existing connections, and (iii) traffic with the other target addresses will be dropped.

Getting rendezvous address:



- We get the IPv6 address of a domain name using DNS.
- We keep the first 64 bits and replace the last 64 bits using the generated value from the hash function with key, salt, and timestamp.
- The server updates the address every minute. To handle clock drift, the server keeps two addresses at a given time.

Chhoyhopper Implementation

SSH

- We provide a Python script that runs the server by updating ip6tables rules, NAT rules, and interface addresses.

```
bash-4.2$ sudo ./chhoyhopper-server --v --address vm18.ant.isi.edu --keyfile file.bin
chhoyhopper-server on clear: 2001:1878:401::8009:1d15
Internal server at: 2001:1878:401::f
at 2021-11-22 14:33:16.863326 accepting 2001:1878:401::f
at 2021-11-22 14:33:16.894085 accepting 2001:1878:401::6f49:b70a:179f:e544
at 2021-11-22 14:33:16.901609 accepting 2001:1878:401::3d57:7162:004f:cc36
at 2021-11-22 14:33:52.082530 accepting 2001:1878:401::6adc:77c5:65f7:9953
at 2021-11-22 14:33:52.087276 dropping 2001:1878:401::3d57:7162:004f:cc36
```

Server running Chhoyhopper for SSH

```
[asmrizvi@localhost client]$ ./chhoyhopper-client --address vm18.ant.isi.edu --keyfile
file.bin
chhoyhopper-client for clear 2001:1878:401::8009:1d15
at 2021-11-22 14:34:03.160524 using 2001:1878:401::6adc:77c5:65f7:9953
The authenticity of host '2001:1878:401:0:6adc:77c5:65f7:9953 (2001:1878:401:0:6adc:77c
5:65f7:9953)' can't be established.
ECDSA key fingerprint is SHA256:pL8CyAxm2m/UqVYhjJ4abX17jvjSKJK5JT1+cFEhjIY.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '2001:1878:401:0:6adc:77c5:65f7:9953' (ECDSA) to the list of
 known hosts.
Last login: Mon Nov 22 14:29:58 2021 from 2001:470:d:1173::1001
############################################################
```

Client connecting to Chhoyhopper SSH server

HTTPS

- For HTTPS, we have **two challenges**.
- **Transparency**: users want service like any other HTTPS service.
  Solution: browser extension to run Chhoyhopper.
- **TLS authentication**: IP-based TLS certificates do not support wildcarding and a static DNS name would reveal the destination.
  Solution: TLS certificate for a wildcard domain name, then dynamically create hopping domain name under that wildcard. Dynamic DNS maps hopping name to the changing IPv6 address.
- We currently provide this extension for Mozilla Firefox.

```
chhoyhopper-server on clear: 2804:269c:8::f
Internal server at: 2804:269c:8::f
Added DNS entry for 0967cf2e75455c692ff01e1bf1e26b2aba7b8718.hop.ant.isi.edu-> 2804:269c:8::0967:cf2e:7545:5c69
Added DNS entry for e73402cc8c0699521eda56c9914d892e4784582a.hop.ant.isi.edu-> 2804:269c:8::e734:02cc:8c06:9952
Added DNS entry for 3276cf211b556cf6df3c09bb434ebe6d1d276a2e.hop.ant.isi.edu-> 2804:269c:8::3276:cf21:1b55:6cf6
Added DNS entry for 071153841b45931991cfaa7fb4f3c8a755d2b5fa.hop.ant.isi.edu-> 2804:269c:8::0711:5384:1b45:9319
```

Server running Chhoyhopper for HTTPS

Browser extension redirects client to the current domain name

Risk of Discover and Collisions

Risks of discover and collisions are tiny.

Discovery: not in our lifetime! Scanning at 100 Gb/s, expected time to discover one server in one /64 is 3000 years.

Collision: it takes a million servers to get a collision in 70 years. Collisions are like the birthday problem, but in a "year" with 2^64 days. The probability of collision is $1 - e^{\frac{-k(k-1)}{2N}}$, with N = 2^64 for k servers.

Conclusions

- We show the design and implementation of Chhoyhopper.
- We plan to provide a Chhoyhopper client as a patch to OpenSSH and provide HTTPS support for Chrome.
- Our implementation for SSH and HTTPS applications is freely available at: https://ant.isi.edu/software/chhoyhopper/.

Chhoyhopper: A Moving Target Defense with IPv6
Research support by DARPA under Contract No. HR001120C0157
ACSAC 2021 Virtual Poster Session / December 2021