Looking and Peering

Attacking from beyond BGP adjacency

WHOAMI

Bo-Shiun Yen

@Bronson113

Security Engineer @ Calif



October 2021

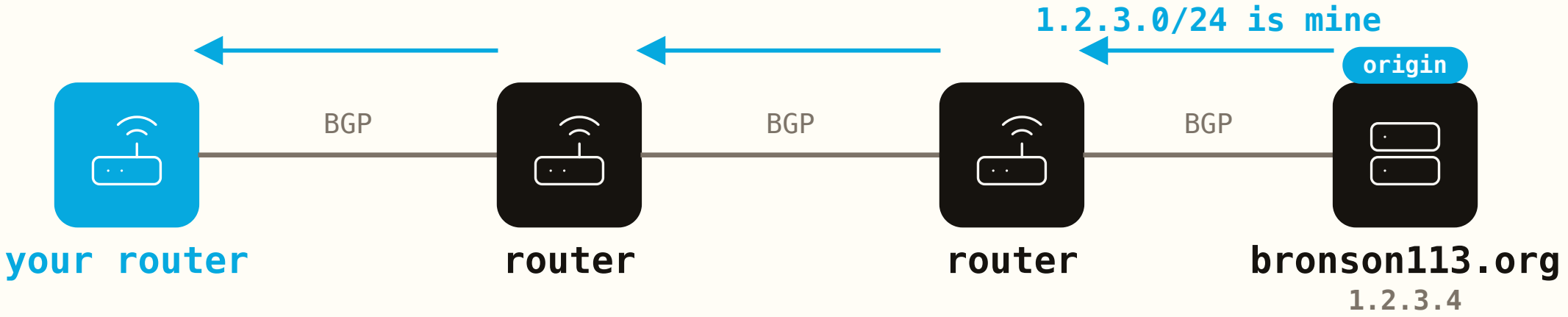
Facebook misconfigured BGP

Facebook disconnected

PART 0

What is BGP

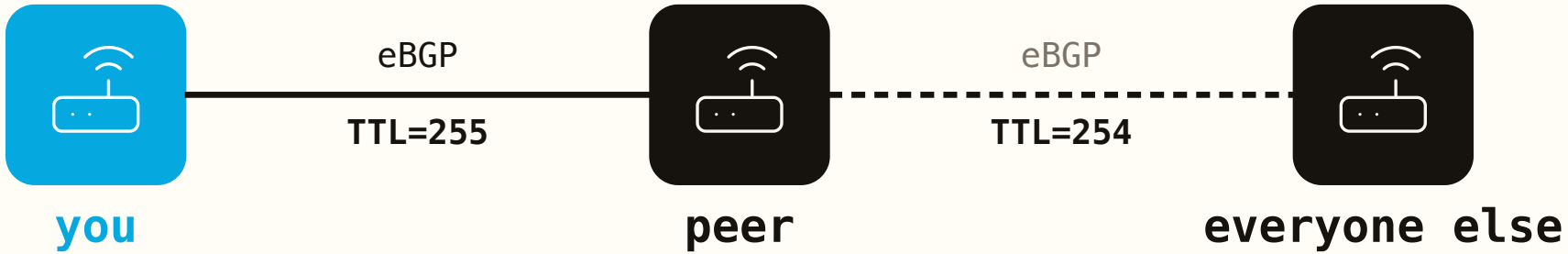
BGP - BORDER GATEWAY PROTOCOL



How are they secured?

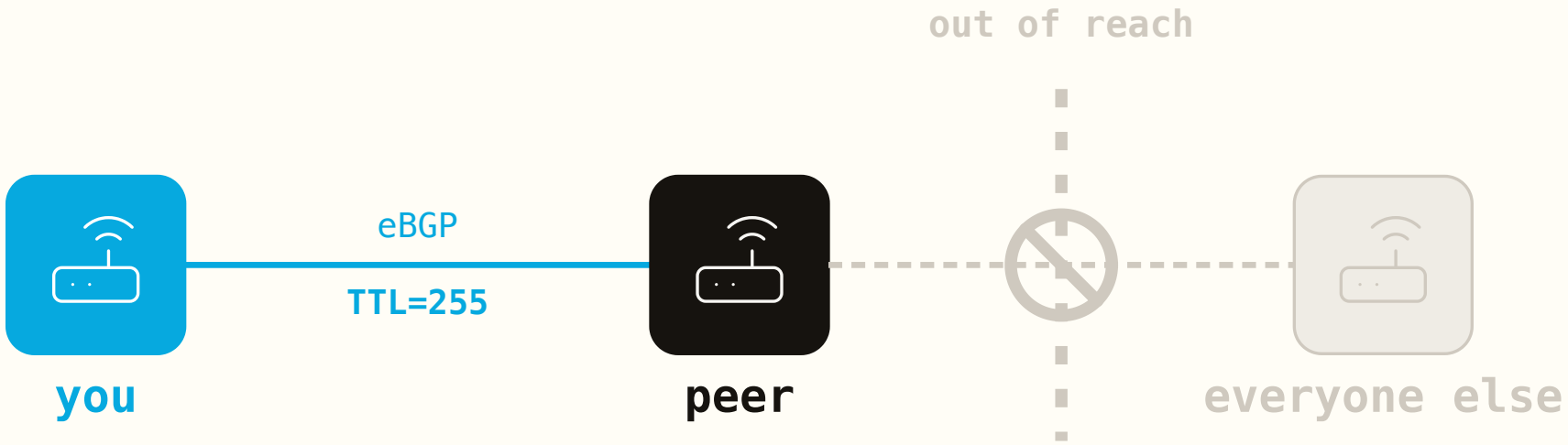
internal vs external

GENERALIZED TTL SECURITY MECHANISM (GTSM)



eBGP accepts UPDATEs
But only if TTL == 255

EVERYTHING LIMITED TO ONE HOP



blast radius = direct peers

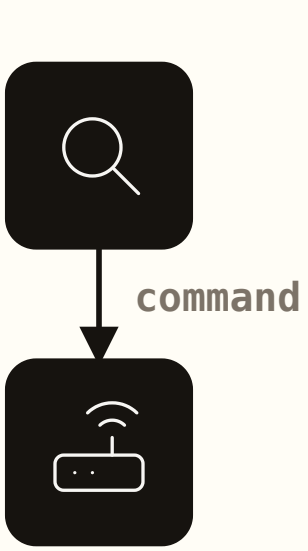
IMPLEMENTATION	WHERE
Closed Source	Majority - Palo Alto, Juniper, Cisco...
BIRD	Common in IXPs and ASNs
FRR bgpd	Common everywhere else
OpenBGPD	Less Common

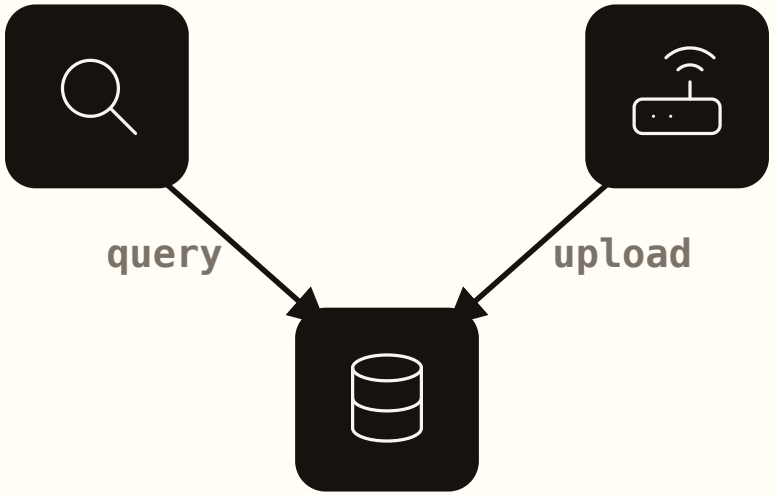
PART 1

Looking glasses

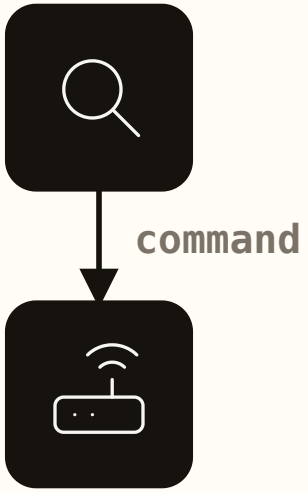
How to debug routing with your peer?
Through Looking Glasses

HOW DOES LOOKING GLASS WORK?





HOW DOES LOOKING GLASS WORK?



Command injection

> Through the Looking-Glass, and What Eve Found There
>
> DEF CON 22 - Luca Bruno, Mariano Graziano - 2014

Two new command injections in 2026: Hyperglass & looking-glass

They're on PeeringDB

IMPLEMENTATION	ASNS	REACHABLE PEERS
Hyperglass	~100 ASNs	~450 peers
looking-glass	~200 ASNs	~400 peers

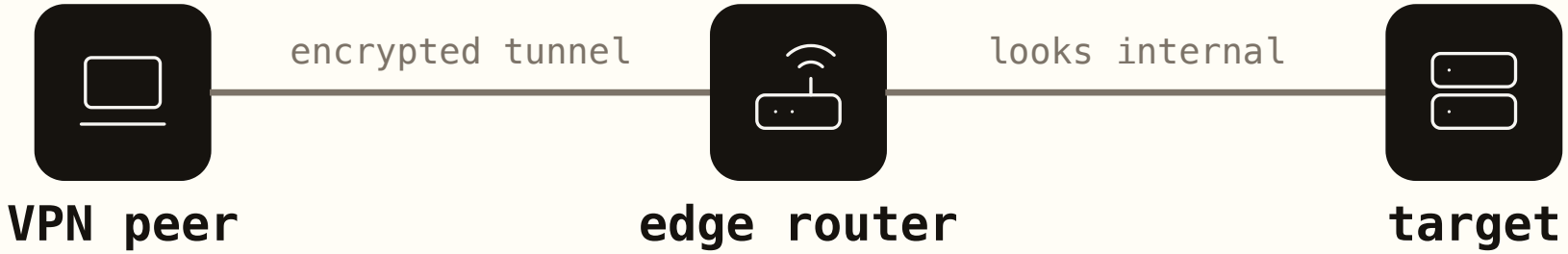
measured via PeeringDB + querying looking glasses

PART 2

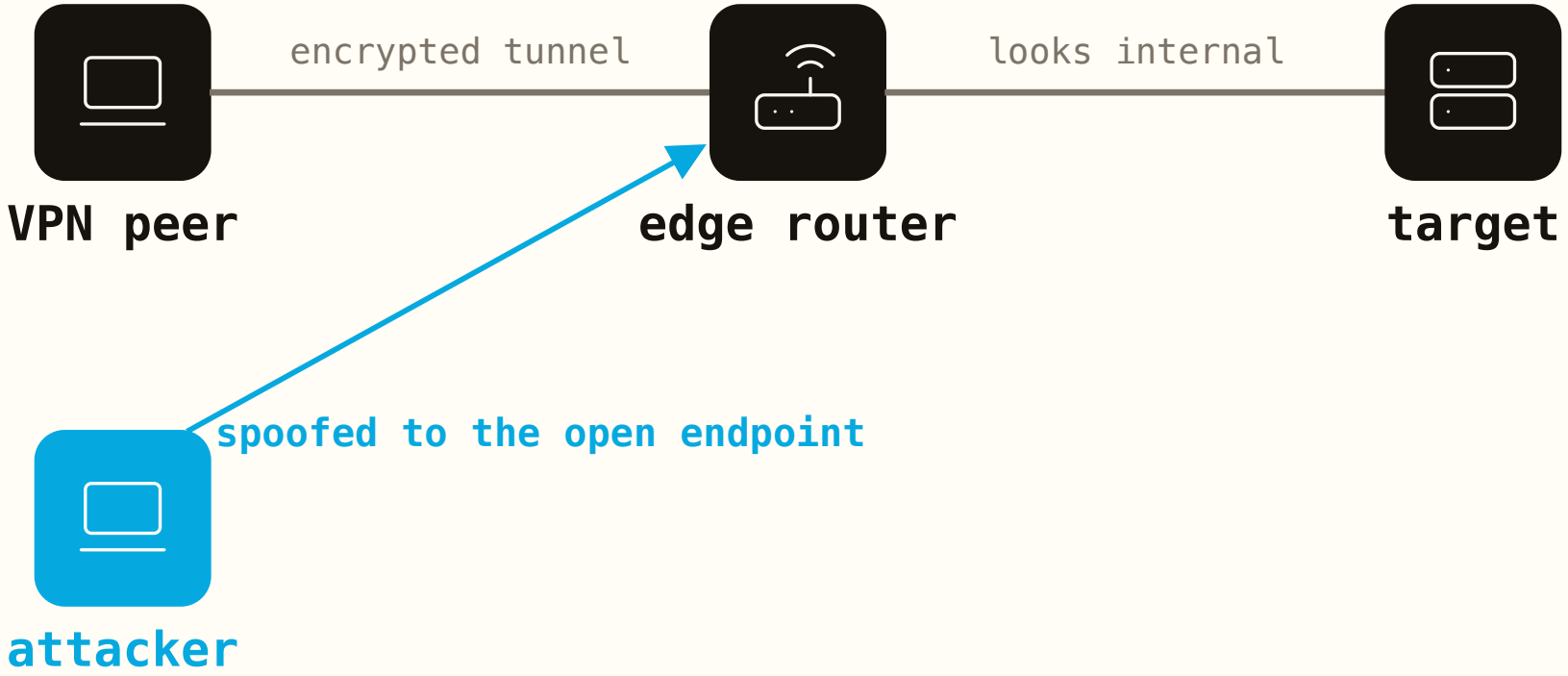
Tunnel injection

> From Spoofing to Tunneling: New Red Team's Networking Techniques for Initial Access and Evasion
>
> DEF CON 33 - 123ojp - 2025

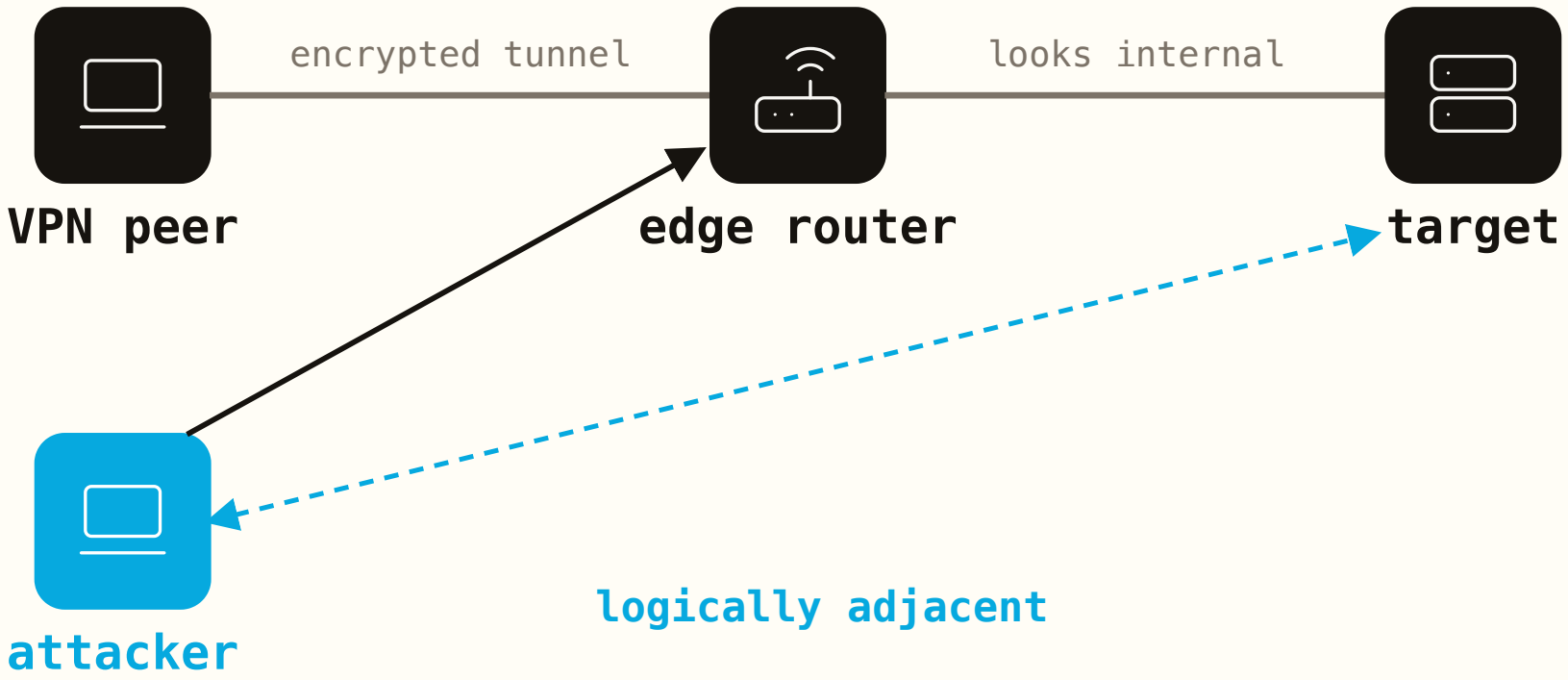
WHAT'S TUNNEL INJECTION?



WHAT'S TUNNEL INJECTION?



USE THE EDGE ROUTER AS OUR HOP



CASE STUDY

UAF in encap intern logic

FRR 10.6.0 and later

BGP_ATTR_INTERN()

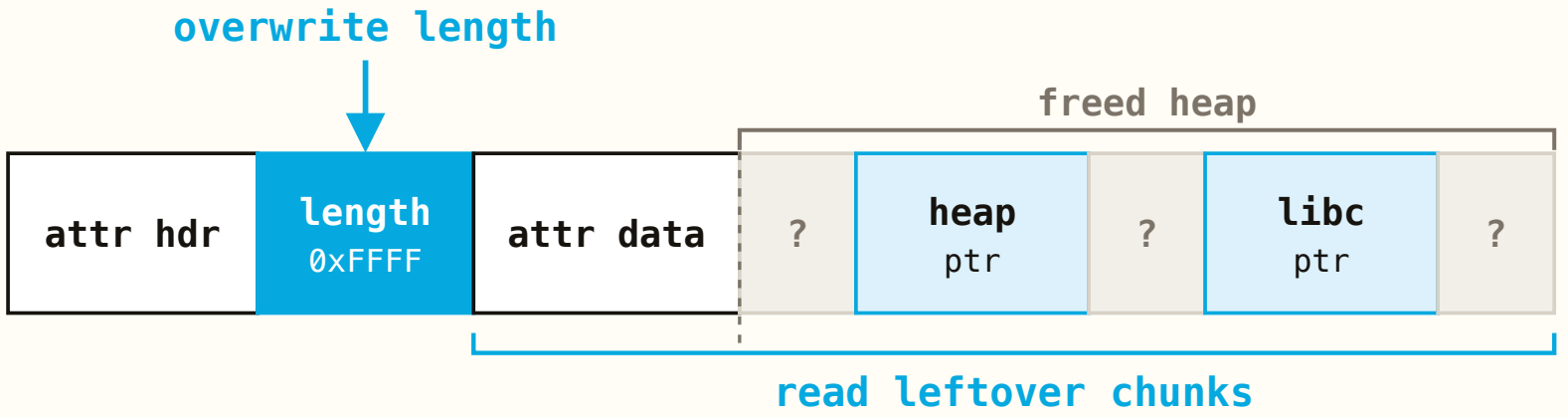
```
// intern the attr, then cache it for the next prefix
find = hash_get(attrhash, attr, bgp_attr_hash_alloc);

if (attrhash_cmp(attr, reuse_anchor->...parsed_attr)) // match attr
    reuse_anchor->...interned = find;   // cached find
```

HASH_GET() -> BGP_ATTR_HASH_ALLOC()

```
// hash_get() calls this allocator on a cache miss
static void *bgp_attr_hash_alloc(void *p)
{
    struct attr *val = p;                    // p == OUR attr
    struct attr *attr = XMALLOC(MTYPE_ATTR, sizeof(*attr));
    *attr = *val;                            // copy into the hashed attr
    if (val->encap_subtlvs)
        val->encap_subtlvs = NULL;           // <- mutates OUR attr
    return attr;
}
```

Side effect on input attr -> attr != find

PACKET 1



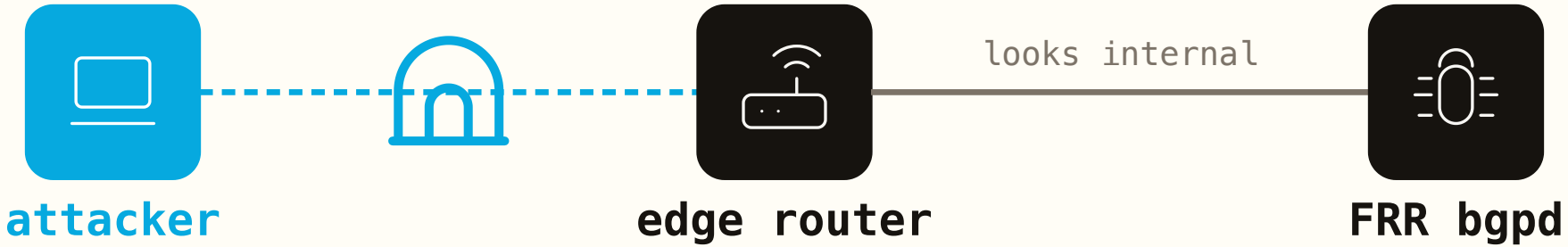
whatever heap + libc pointers were there -> ASLR leak

PACKET 2



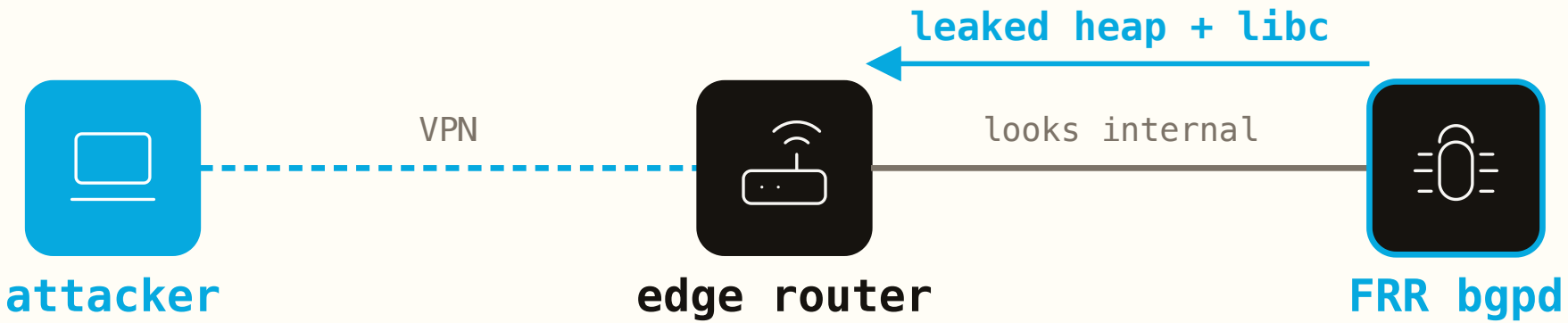
the next hash lookup calls hash_cmp() -> RCE

COMBINE WITH TUNNEL INJECTION



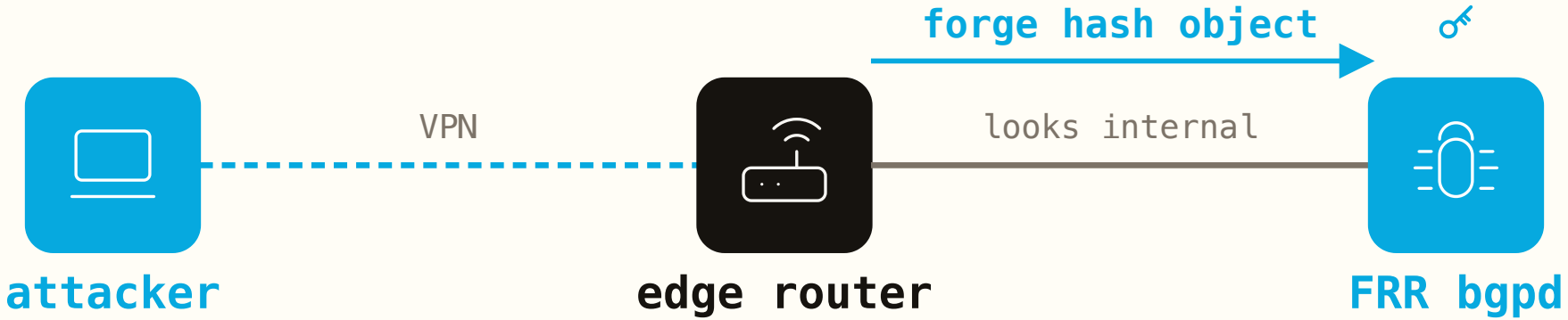
a BGP UPDATE wrapped in a tunnel

COMBINE WITH TUNNEL INJECTION



packet 1: heap over-reads

COMBINE WITH TUNNEL INJECTION



packet 2: RCE

IMPACT SO FAR

One hop

The next-hop boundary means we reach direct peers only

PART 3

Transitive BGP attacks

Variant 1:

Core Parser Issues

CVE-2026-49943

BIRD as_path_match stack overflow

THE BUG - `AS_PATH_MATCH()`

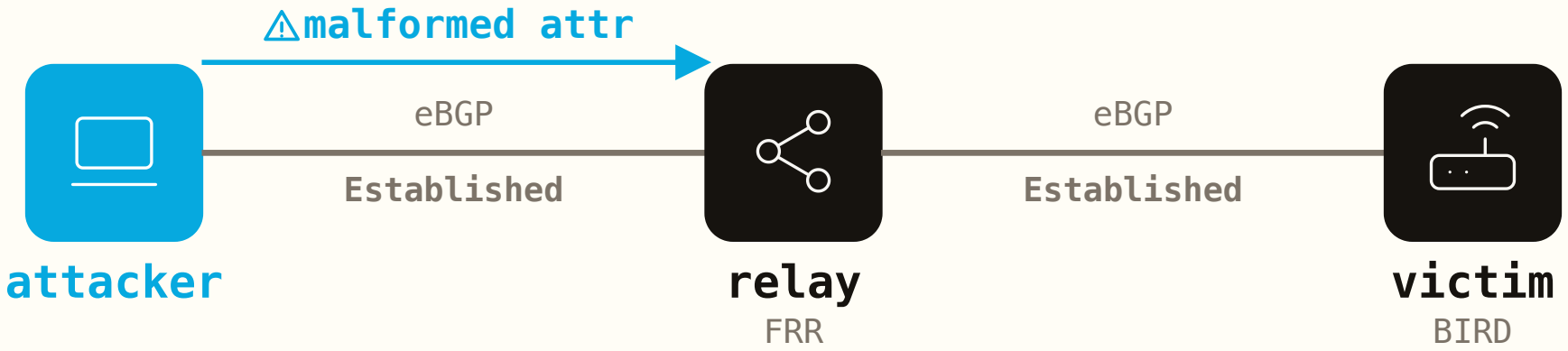
```
int as_path_match(const struct adata *path, ...)
{
    struct pm_pos pos[2048 + 1];          // fixed ~32 KB stack array
    int plen = parse_path(path, pos);     // one 16-byte write per ASN - no bound
    ...
}
```

forge long AS_PATH with ext-msg -> stack overflow pos[]

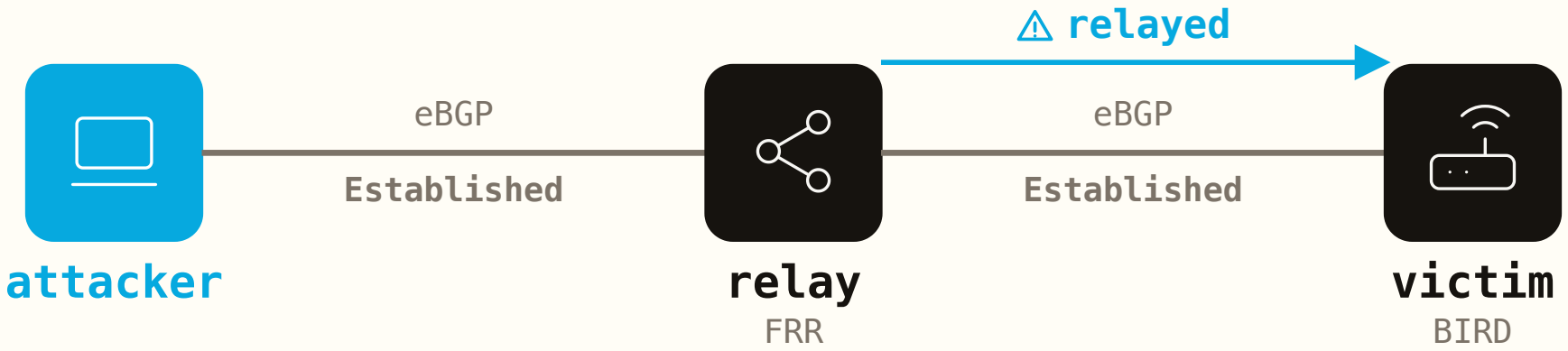
TRANSITIVE ATTACKS



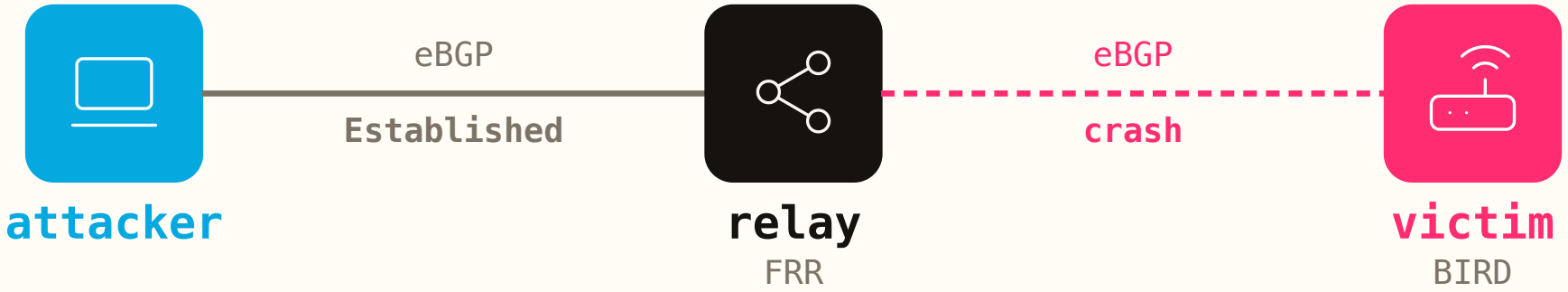
TRANSITIVE ATTACKS



TRANSITIVE ATTACKS

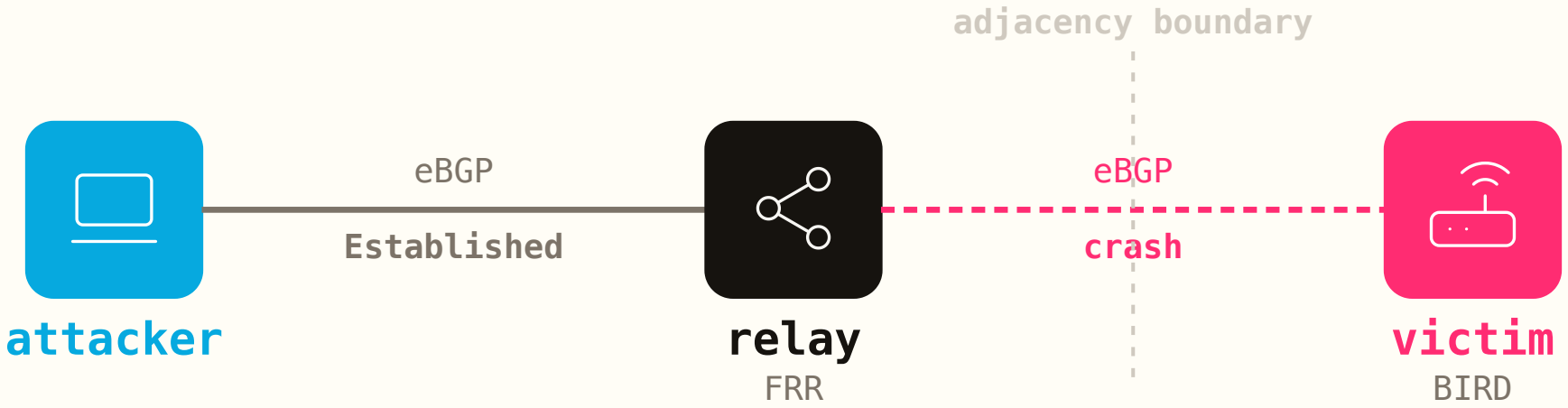


TRANSITIVE ATTACKS



the parser crashes
stack overflow

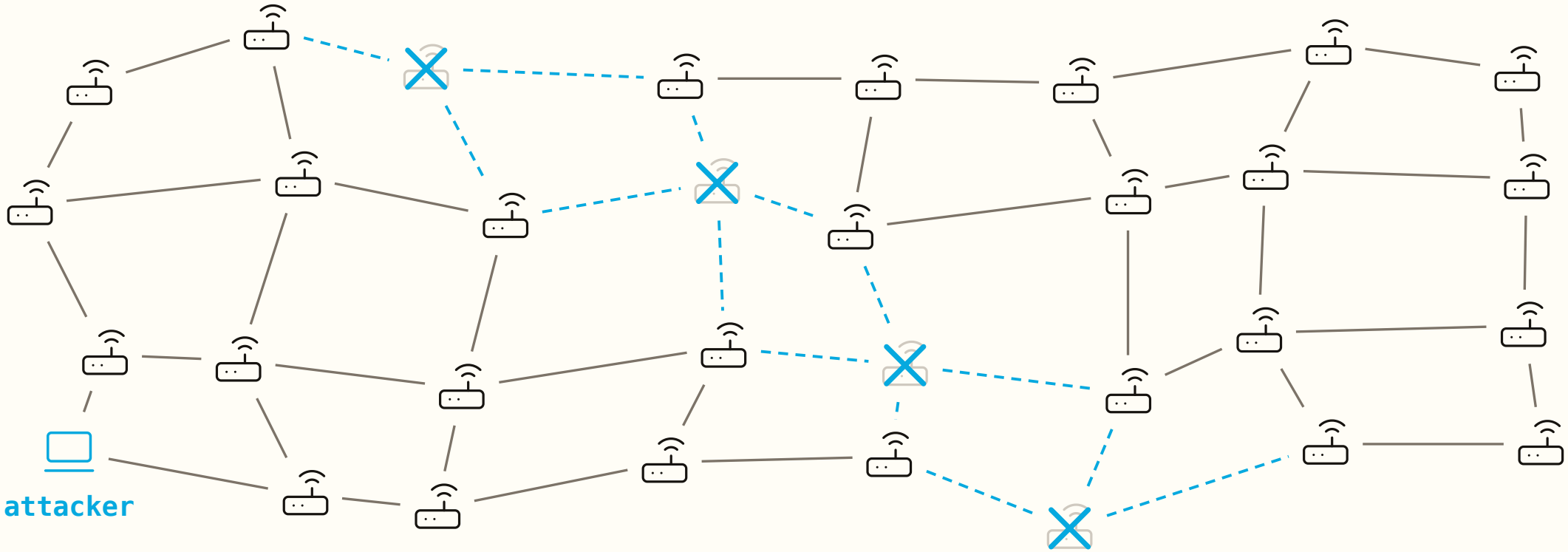
TRANSITIVE ATTACKS



the victim is not a peer to the attacker
the attribute crosses the security boundary

SEGMENTING THE INTERNET

BIRD route servers – all crash



the whole BIRD border crashes -> the internet splits in two

Variant 2:

Optional Attributes

> "...the unrecognized transitive optional attribute of that path **MUST be passed**, along with the path, to other BGP peers..."
>
> RFC 4271

Unknown transitive attributes get re-emitted

RFC 4271 ALSO SAID

Malformed attribute

-> session reset

This is a bad behavior

> one malformed BGP-LS attribute reset Juniper's sessions - riding transitively past every router that didn't parse it
>
> CVE-2023-4481 - benjojo, "Grave flaws in BGP Error handling" - 2023

RFC 7606 recommends treat-as-withdrawal

Now let's consider parser issue again

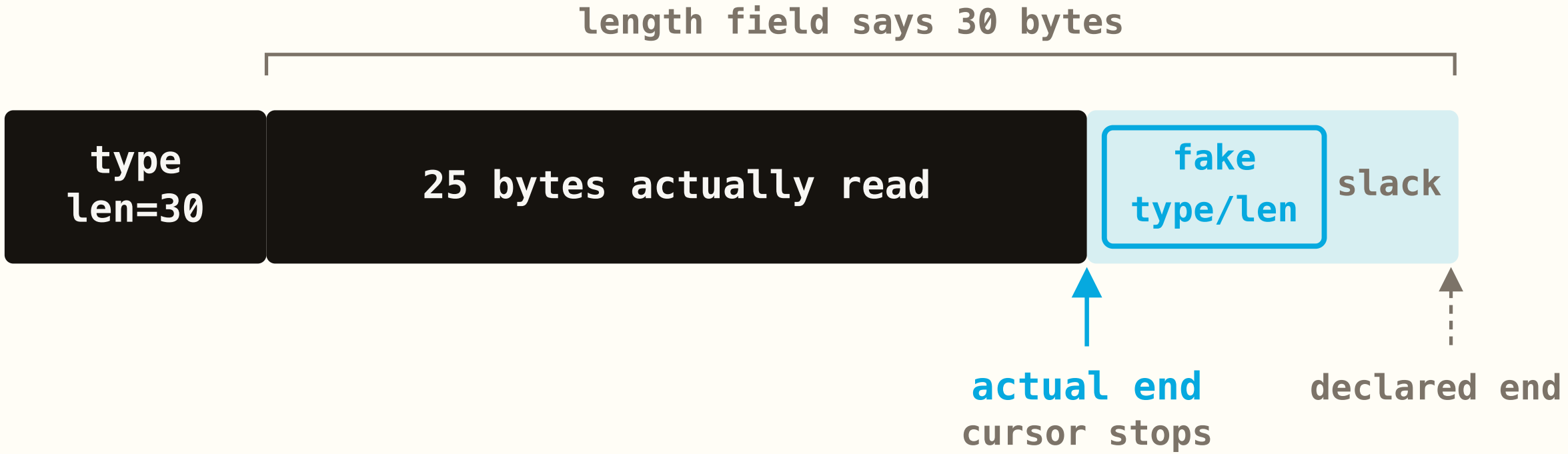
FRR Prefix-SID length desync

THE BUG - `BGP_ATTR_PREFIX_SID()`

```
type   = stream_getc(connection->curr);
length = stream_getw(connection->curr);

ret = bgp_attr_psid_sub(type, length, args);   // may read fewer than `length`
psid_parsed_length += length + headersz;       // bookkeeping trusts the declared length
```

the cursor and the counter disagree -> it lands off the attribute boundary

THE LENGTH DESYNC



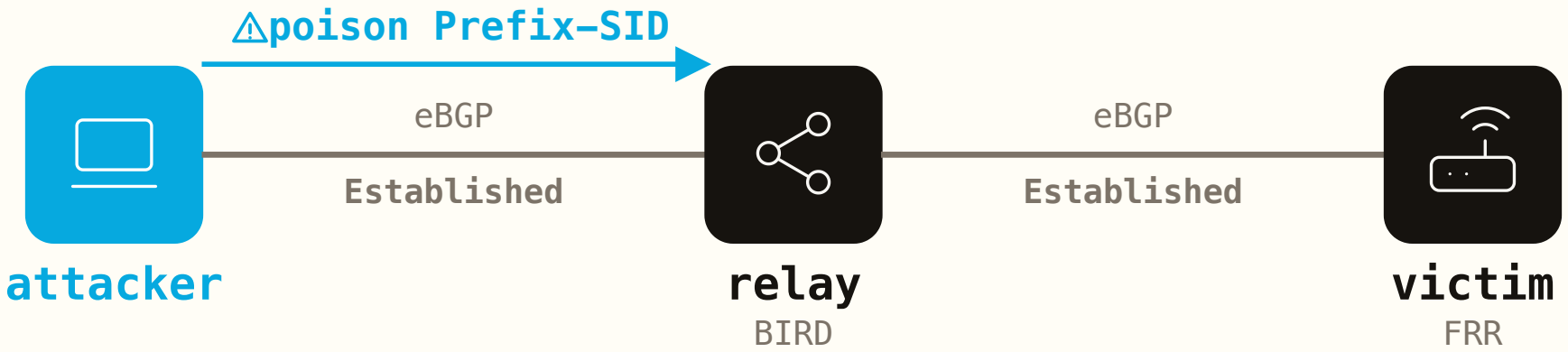
Unparsable slack -> session reset

Crafted slack -> packet smuggling

WHAT ATTRIBUTE IS THIS

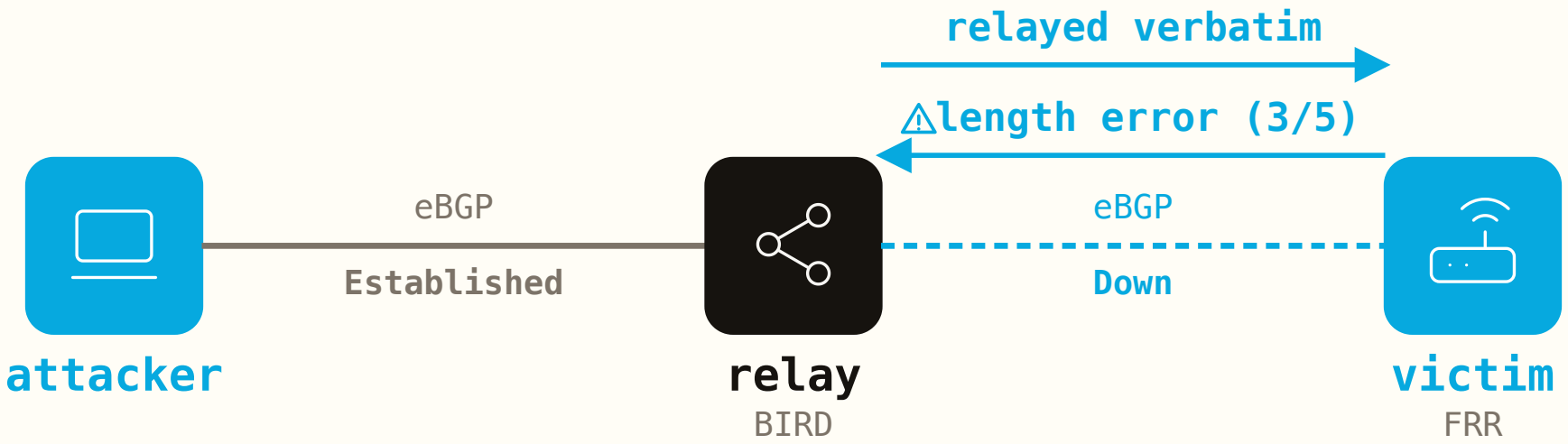
Prefix-SID: Optional Transitive

TRANSITIVE ATTACK 2



optional-transitive attribute

TRANSITIVE ATTACK 2



cursor lands off the attribute boundary
session reset

no adjacency needed (again)

Variant 3:

Configuration Difference

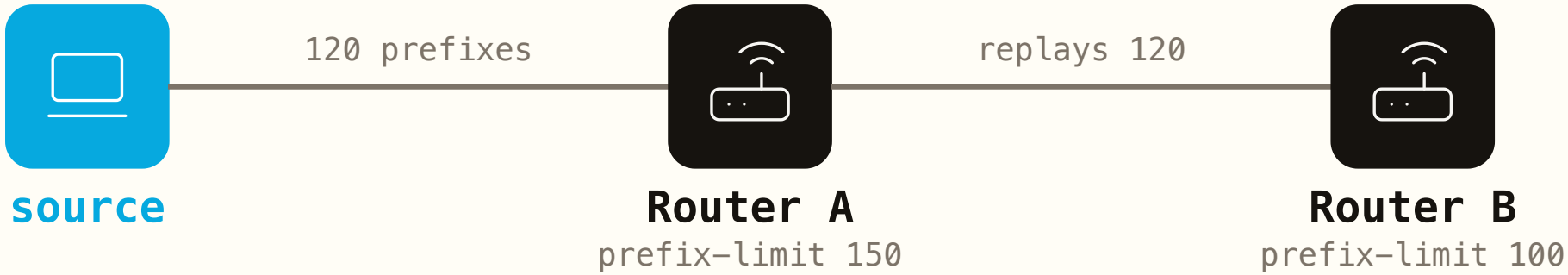
What if the attribute is well formed
But routers disagree with each other?

THE PREFIX LIMIT

```
BIRD          import limit 100 action restart
FRR / Cisco   neighbor X maximum-prefix 100 restart 5
```

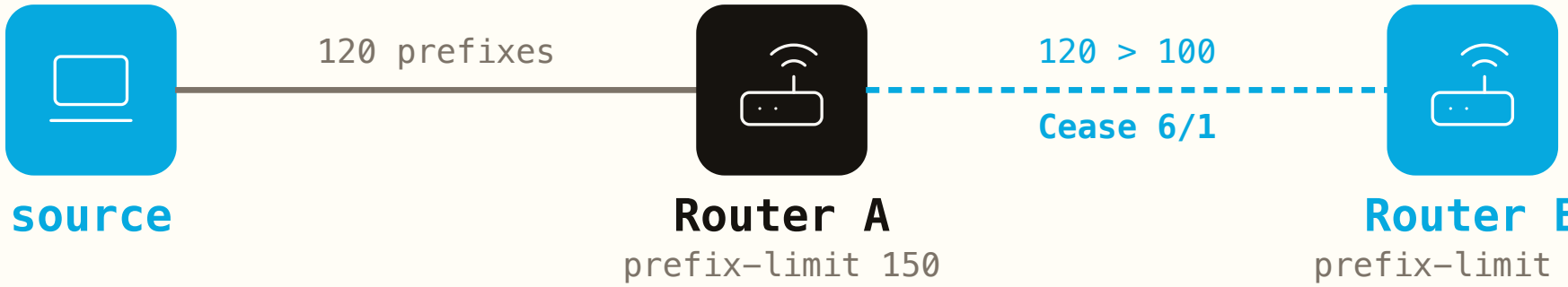
limit how many prefixes a peer may send

LIMIT MISMATCH



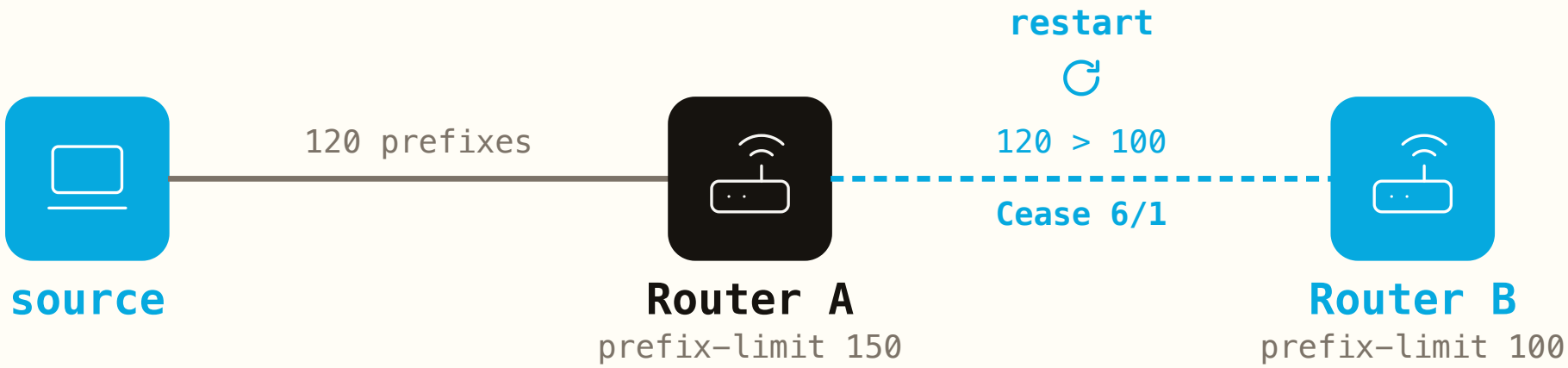
A accepts (120 < 150)
and replays all 120 to B

LIMIT MISMATCH



B rejects (120 > 100)
Cease 6/1: Maximum Number of Prefixes

LIMIT MISMATCH

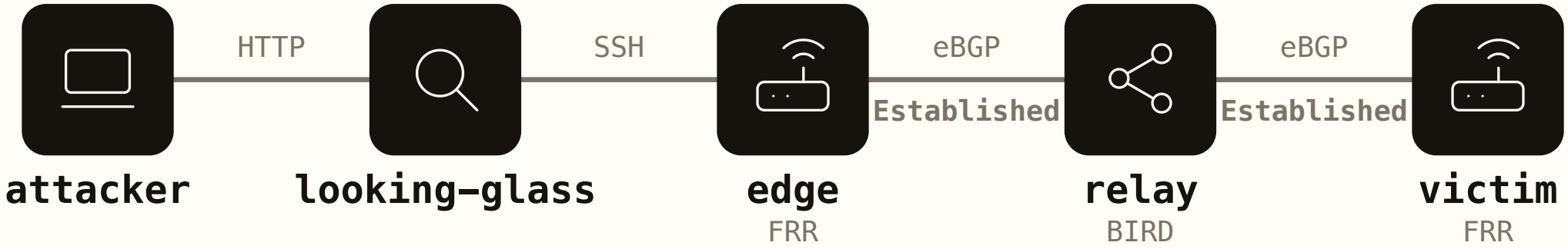


B restarts, but A still holds the routes
persistent reset loop

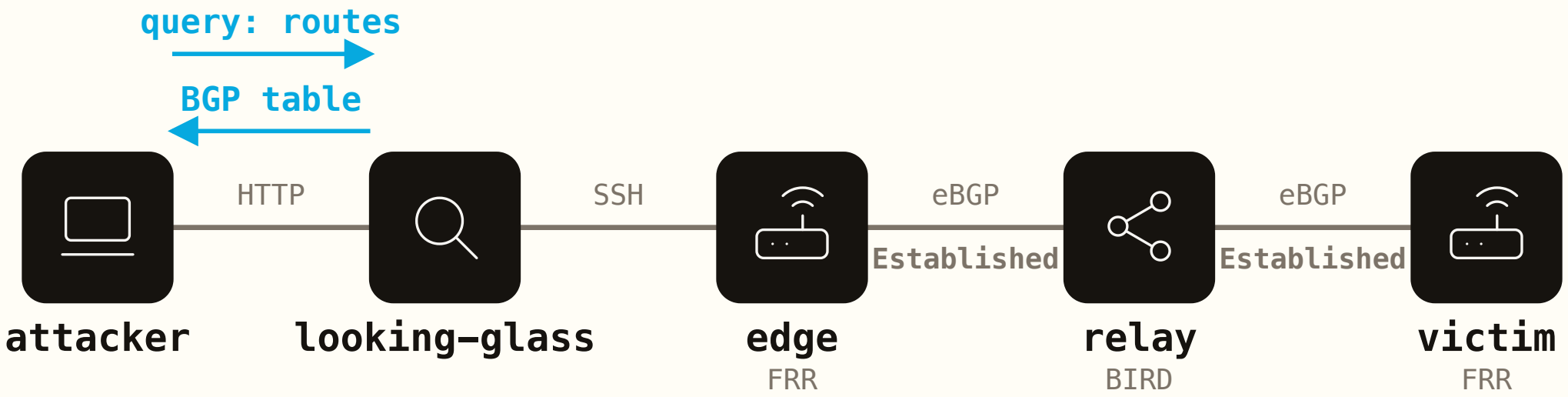
PART 4

Demo

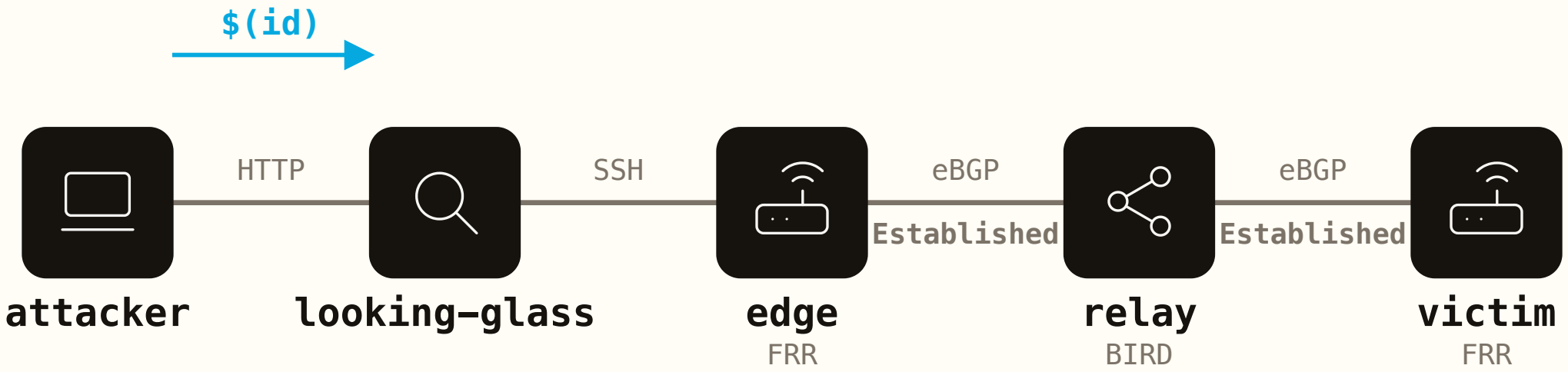
THE LAB


HTTP
SSH
eBGP
Established
eBGP
Established
attacker
looking-glass
edge
FRR
relay
BIRD
victim
FRR

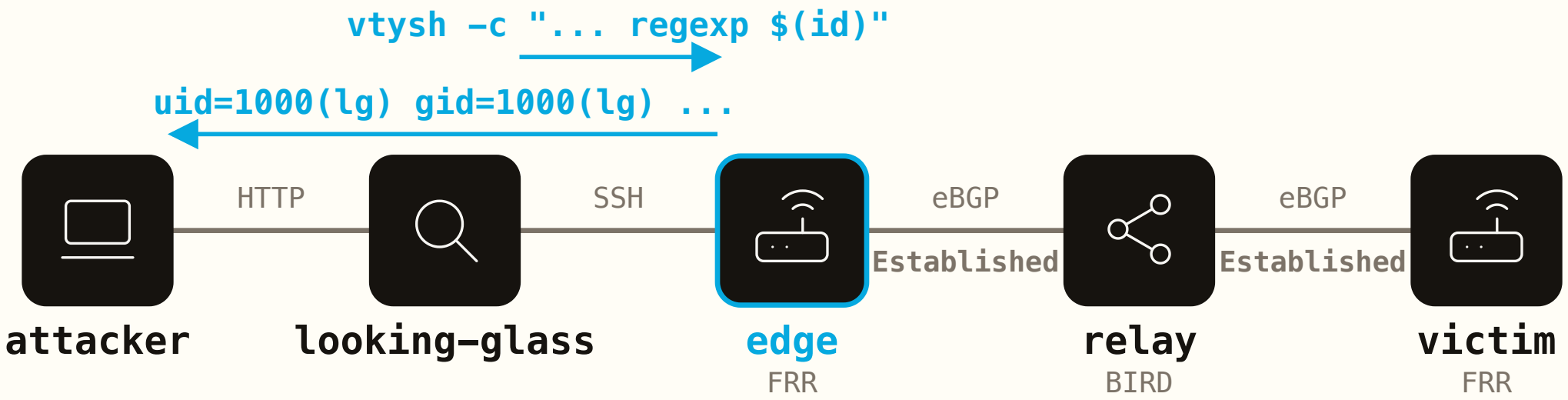
STAGE 0 - RECON



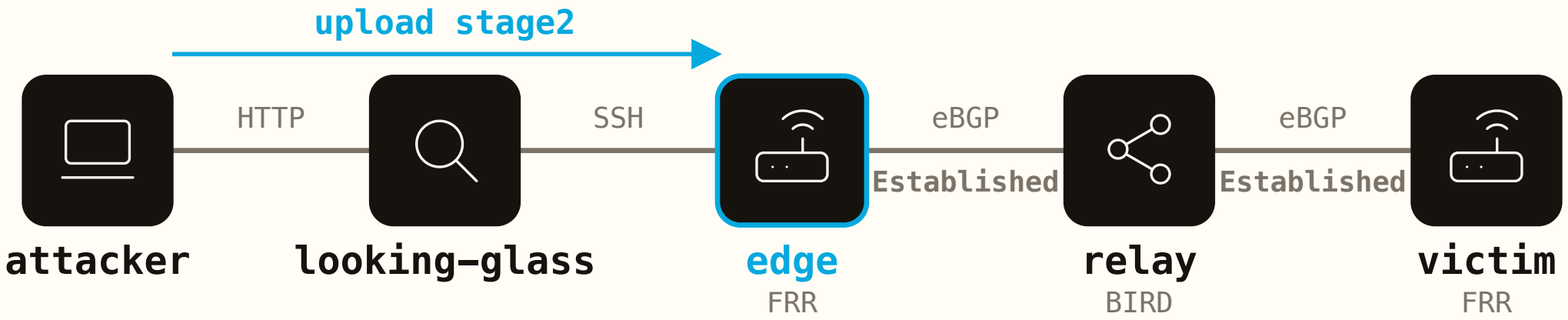
STAGE 1 - RCE


$(id)
HTTP
SSH
eBGP
Established
eBGP
Established
attacker
looking-glass
edge
FRR
relay
BIRD
victim
FRR

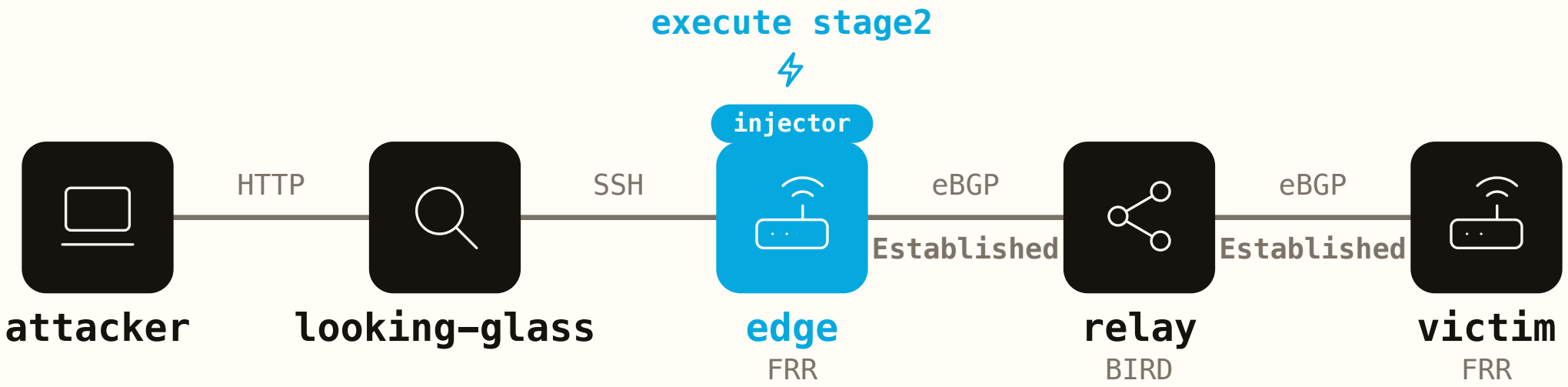
STAGE 1 - RCE


vtysh -c "... regexp $(id)"
uid=1000(lg) gid=1000(lg) ...
HTTP
SSH
eBGP
Established
eBGP
Established
attacker
looking-glass
edge
FRR
relay
BIRD
victim
FRR

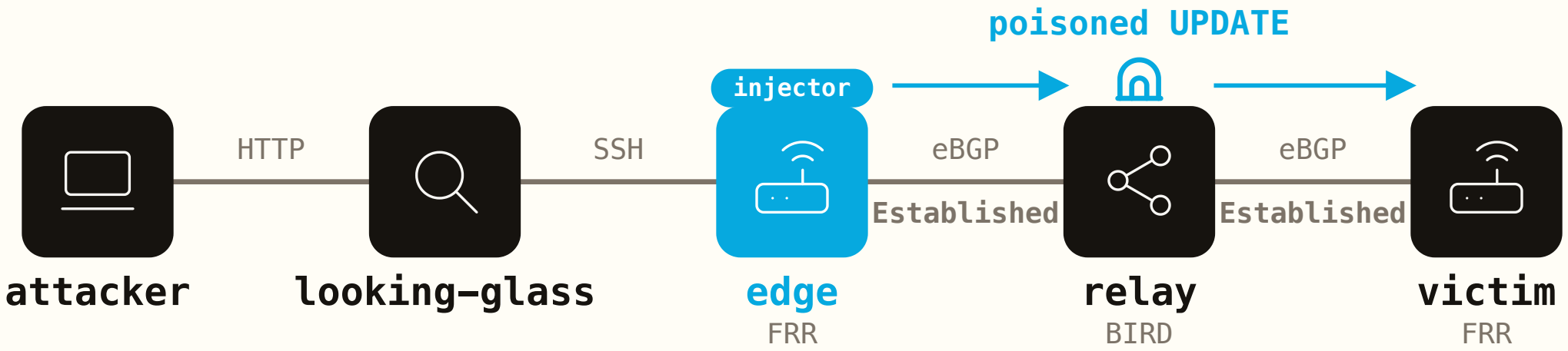
STAGE 2 - PIVOT



STAGE 2 - PIVOT

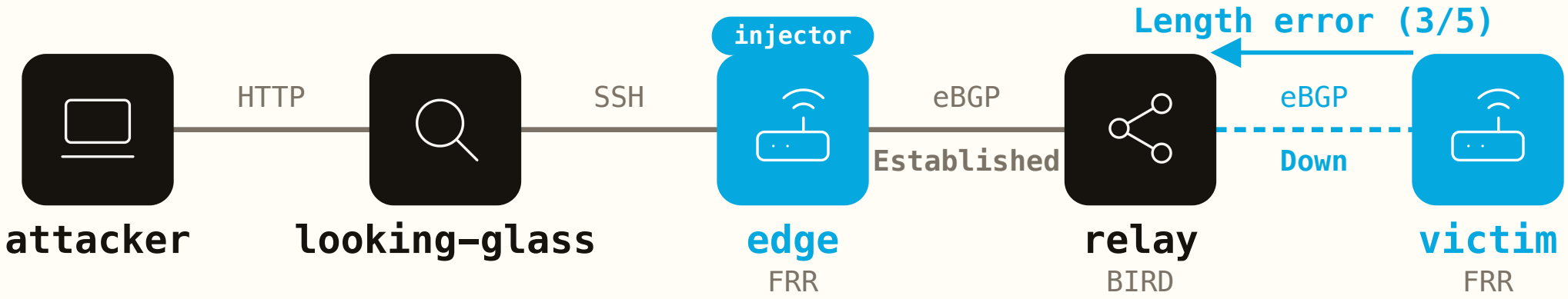


STAGE 3 - REMOTE DOS

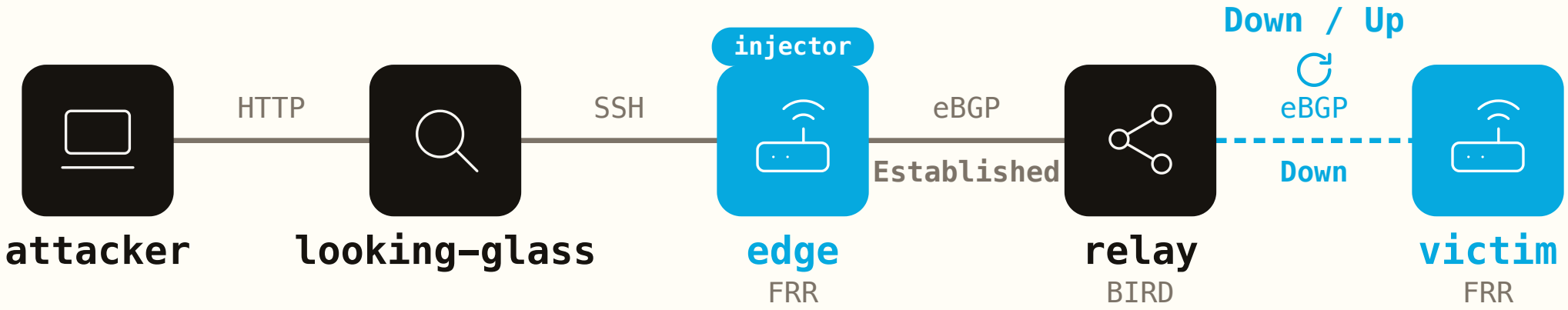


Prefix-SID attack

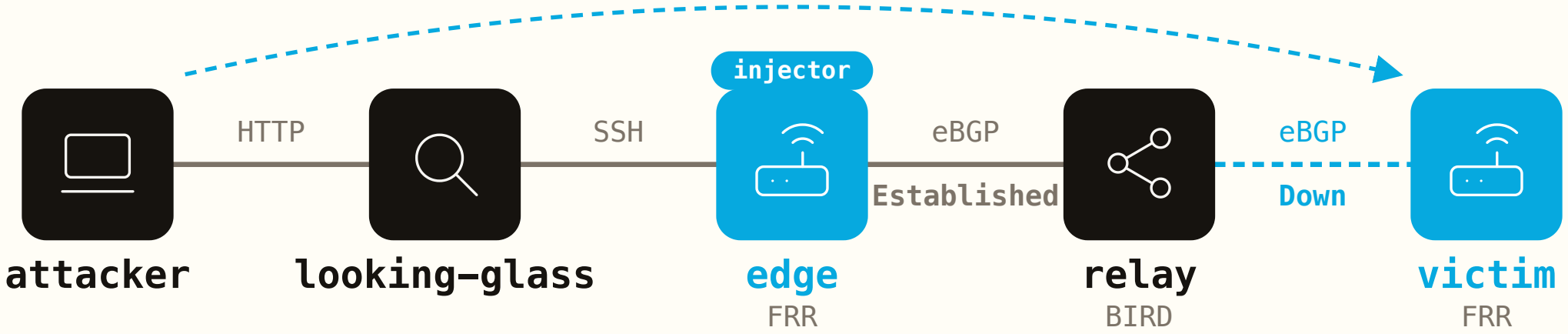
STAGE 3 - REMOTE DOS



STAGE 3 - REMOTE DOS



ATTACK BEYOND ADJACENCY



Play the demo

```
── EXPLOIT - ./demo.sh ──────────────────────────────────────────────────────

 routers streaming on the left.
 drive the demo here:

   ./demo.sh             step-gated (ENTER between stages)
   ./demo.sh --record    auto + pcaps/logs to ./evidence/
                                                                      /0.1s
➜  demo-chain git:(main) ✗ ./demo.sh                  [17/07/26 | 23:37:19]
```

PART 5

Impact

We fingerprinted published daemon version through ASN information

ESTIMATE SHARE OF EBGP ROUTERS

DAEMON	IXP/ASNS	WORLD WIDE
BIRD	50%	8%
FRR	15%	10%



Thank you