CentralNic Group PLC

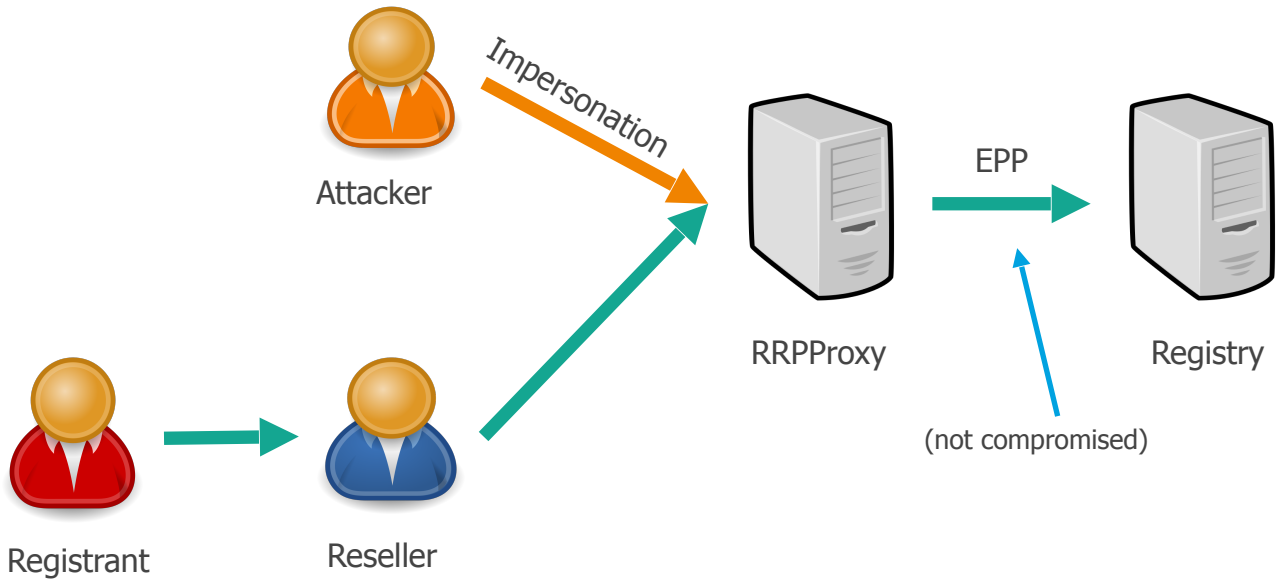
DNSpionage and Registry Lock

Gavin Brown, Chief Innovation Officer
<gavin.brown@centralnic.com>

ROW #8, Bangkok, May 2019

CentralNic's view on DNSpionage

- Attackers targeted domains hosted on PCH.NET and DNSNODE.NET nameservers
- Both domain names were registered through a reseller of Key-Systems*
- Attackers obtained that reseller's credentials and modified nameservers to divert traffic



* Key-Systems is a subsidiary of CentralNic Group plc .

CentralNic Group PLC

CentralNic's view on DNSpionage

- RRPProxy could not distinguish attacker from the reseller they impersonated
- IP whitelisting available, but not configured on reseller account
- 2FA available, but not configured on reseller account
- Registry Lock available, but not configured on what are critical infrastructure domains

Other Potential Victims

- Several other large DNS operator domains were vulnerable, and were contacted shortly after the DNSpionage incident occurred and strongly encouraged to enable Registry Lock

- First response: "we're fine, we have `clientTransferProhibited`"

- After further explanation, agreed to add Registry Lock through their registrar

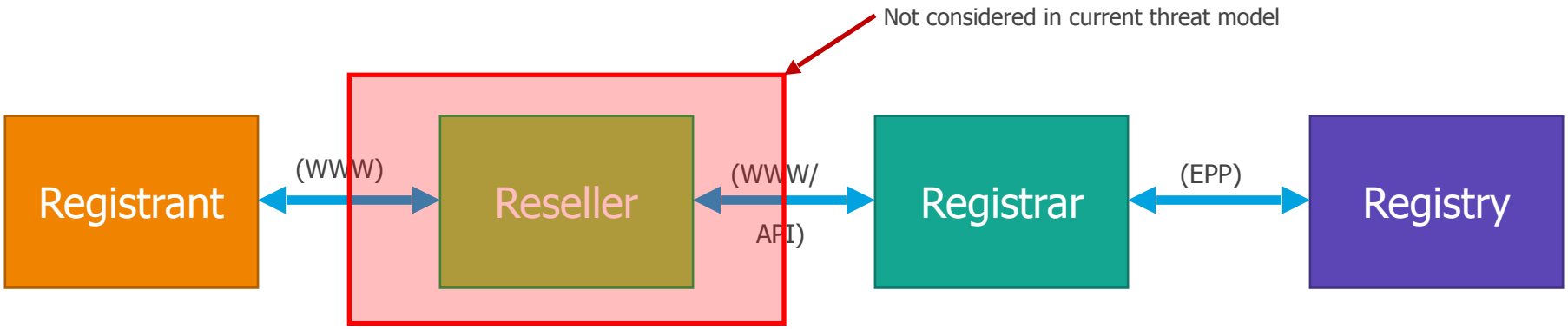
- Three months after the initial contact, none of the operators contacted have managed to add Registry Lock to their domains!

"Three Rs" Model

- This is the actor model for all gTLDs (and many ccTLDs), enshrined in registry agreements, consensus policies and standards documents
- Forms the basis of threat model for (gTLD) domain name industry
- Registrar ⇔ Registry: access control and authn/authz prescribed by registry agreements and standards docs. MFA (ID/PW, TLS cert, IP whitelisting) explicitly required (in RFC 5734, §8)
- Registrant ⇔ Registrar: deregulated, in the hope the market will encourage good practice (registrants will transfer their domains from less secure registrars to more secure registrars)

"Three Rs" Model

- Registrants have poor information about their registrar's security controls
- Registrants (even industry professionals) are not aware of risks and best practices
- Registrants may not even be aware of who their registrar is, because of resellers
- Resellers are even more opaque on security controls than registrars
- There may be multiple resellers!
- An inaccurate threat model leads to inadequate controls



Registry Lock*

- Follows RRR model – no interaction between registrant and registry

- Registry adds `serverUpdateProhibited`, `serverDeleteProhibited`, `serverTransferProhibited` status codes to domains
  - All administration of domain must then be done out-of-band manually

- Pros:
  - Mitigates impact of compromise of a registrar, or the registrar ⇔ registry channel
  - Some parts (e.g. setup, unlock requests) can be automatable, but ultimately most of the system has to be manual

- Cons:
  - Does nothing about a compromise higher up the chain
  - Expensive and inefficient, deterring adoption

* (as implemented in many gTLDs)

Registry Lock: Option #1: Direct Registrant ⇔ Registry Authentication

- Common in ccTLDs, but not permitted under current ICANN rules
- Therefore not a universal solution (registrants have mixed portfolios)

Registry Lock: Option #2: Multi-Factor Authentication

- Registrant supplies a second factor which is then associated with the domain at the registry, which could be:
  - A TOTP key
  - A passphrase
  - Some other cryptographic token to sign unlock requests

- Pros:
  - Mitigates impact of compromise of a reseller and reseller ⇔ registrar channel, as well as the registrar and the registrar ⇔ registry channel
  - Easily automatable by registries

- Cons:
  - Any factor based on symmetric crypto (e.g. TOTP) can be compromised during the setup
  - Asymmetric crypto (e.g. PGP) is much more error-prone and complex to manage
  - Requires lots of changes to channel protocols, and implementation all the way along the chain
  - If registrant loses authentication factor, their domain is doomed!

Registry Lock: Option #3: DNS-based solution

```
_lock.example.com.  3600  IN  TXT "serverUpdateProhibited"
```

- Registry does a DNS query when it receives an `<update>` command

- Pros:
  - Mitigates impact of compromise of a reseller and reseller ⇔ registrar channel, as well as the registrar and the registrar ⇔ registry channel
  - Easily automatable by registries

- Cons:
  - Registrars/resellers out of the loop if they don't also do the DNS query before sending the `<update>` command: will increase their support costs
  - Weak form of authentication unless domain has DNSSEC
  - Transfers risk to DNS operator
  - DNS operator = registrar/reseller for most domains!

Conclusions

- There probably isn't a single solution that suits everyone
- Registry Lock (as implemented in gTLDs) does not scale (but could be made more efficient)
- Any in-band automated system will be compromised, therefore is not fit for purpose
- For registrars to support it, any system must:
  - Be simple
  - Be reliable
  - Be cheap to support
  - Not bypass them as managers of the domains