UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Universidade Tecnológica Federal do Paraná – Câmpus Pato Branco
DAINF – Departamento Acadêmico de Informática

On Omitting Commits and Committing Omissions:

Preventing Git Metadata Tampering That (Re)introduces Software Vulnerabilities

Santiago Torres-Arias, Justin Cappos (New York University)
Anil Kumar Ammula, Reza Curmola (New Jersey Institute of Technology)

Willian Americano Lopes
walopes23@gmail.com

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

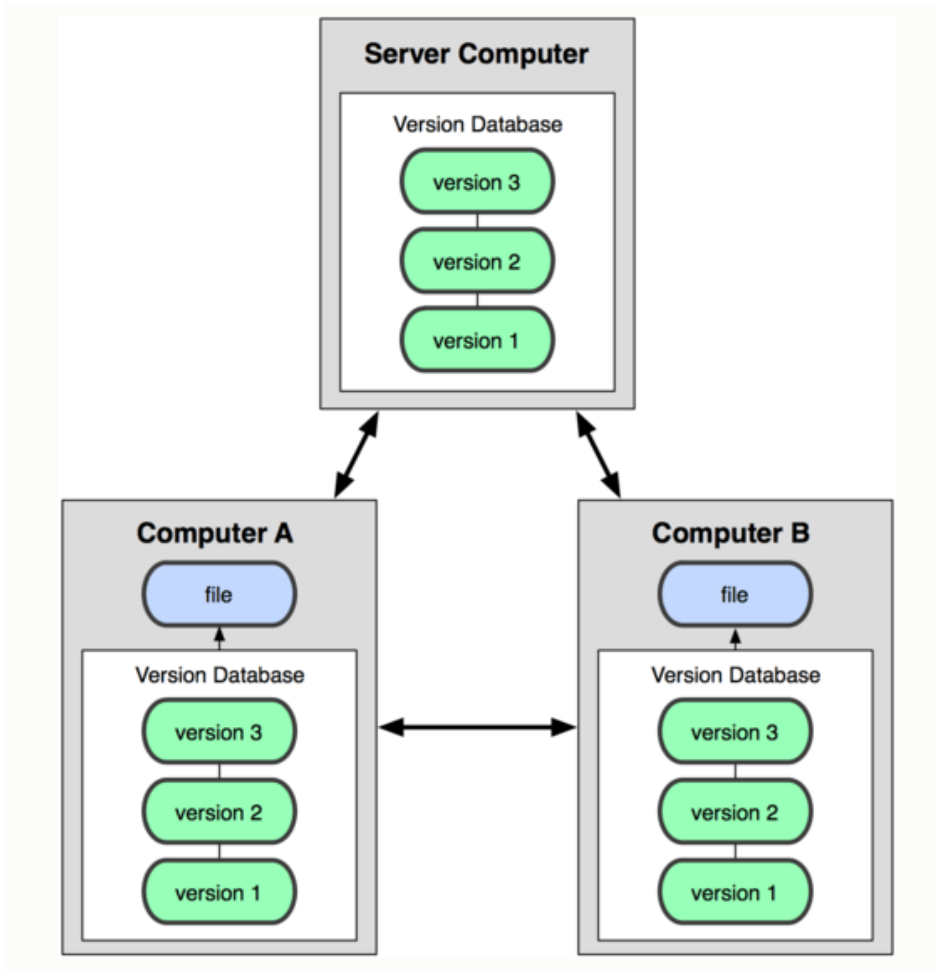
SISTEMA DE CONTROLE DE VERSÃO (SCV)

- Ferramenta indispensável em grandes projetos de *software*
- Desenvolvimento colaborativo
- *Rollback* de versões anteriores
- Mantém um histórico progressivo do projeto

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

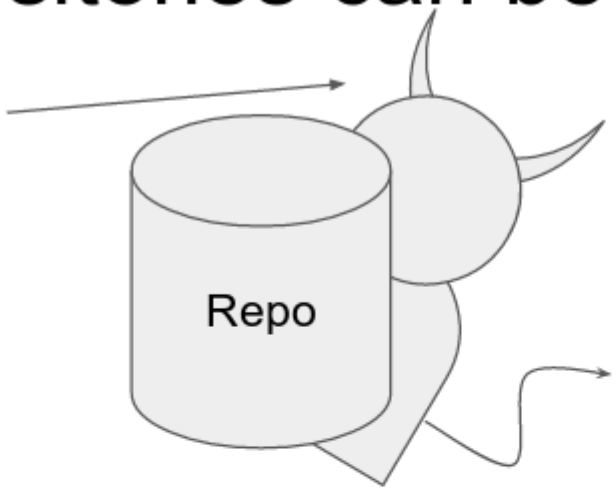
- Sistema de Controle de Versão Distribuído (SCVD)
- Clientes são cópias dos repositórios
- Criado por Linus Torvalds
- Vários repositórios remotos

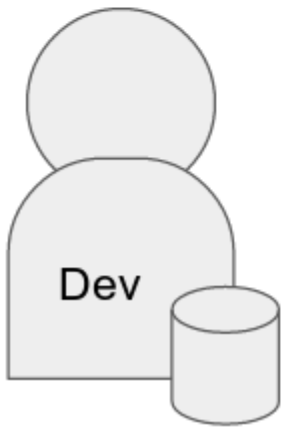
PROBLEMA!

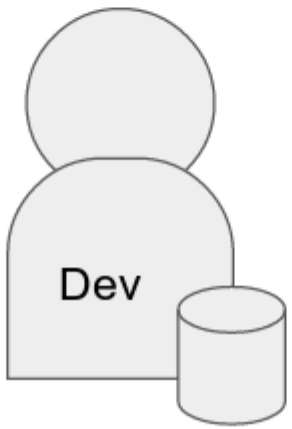


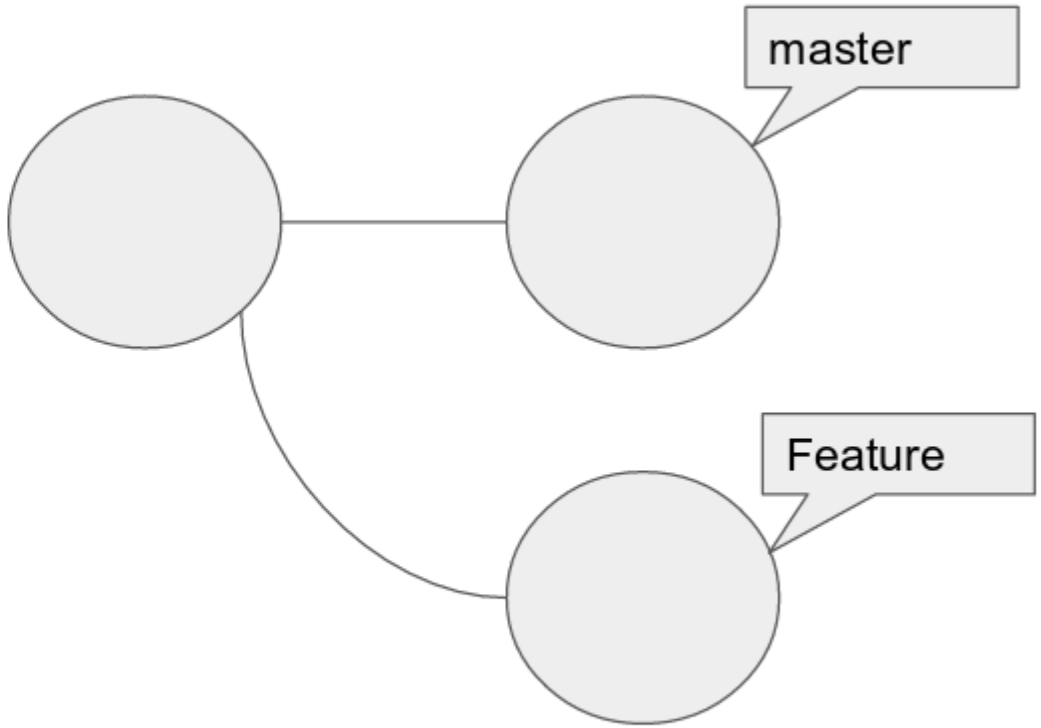
Git repositories can be compromised

Wants to
Watch the
World burn









1

WIRED 'Google' Hackers Had Ability to Alter Source Code

BUSINESS CULTURE DESIGN

SHARE

SHARE

TWEET

PIN

COMMENT 0

EMAIL

KIM ZETTER SECURITY 03.03.10 11:05 PM

'GOOGLE' HACKER ABILITY TO ALTER CODE

https://www.wired.com/2016/09/how-to-steal-an-ai/

GEEK

NEWS

Major Open Source code repository hacked for months, says FSF

BY
08.14.2003 :: 11:01AM EDT

0 SHARES

If you've downloaded any free, Open Source software since March of this year you might've downloaded more than you bargained for. It seems that back in March of 2003 someone compromised the root FTP servers that function as the code repository for thousands of Open Source software projects. The compromise was severe enough that the attacker could have inserted trojaned code into any project he or she desired. GNU.org is recommending that any who downloaded code after February to check that code against a **listing of known good MD5 sums**

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

CONCEITOS DE GIT

- *Commit*: Salvam mudanças no repositório
  Objetos comitados contém metadados (quem e quando foi comitado, commit anterior etc)
- *Branch:* Apontam para os metadados
- *Fetch:* Recupera commits de outro repositório remoto
- *Merge*: Junta as alterações feitas em um histórico
- *Push:* Envia commits de um repositório local para um remoto
- *Pull*
- *Tag:* Marca uma revisão de código específico

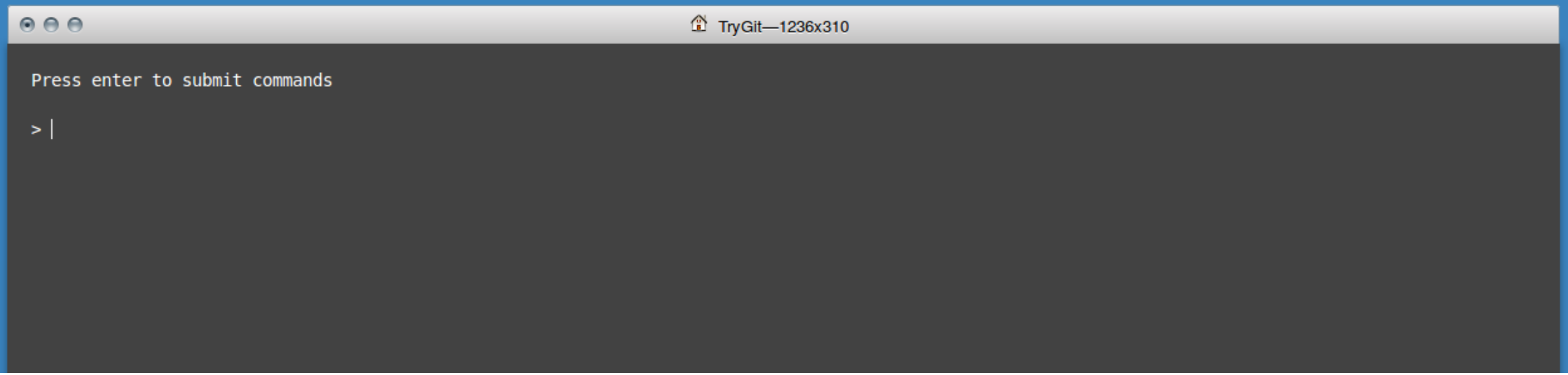
1.1 · Got 15 minutes and want to learn Git?

tryGit

Git allows groups of people to work on the same documents (often code) at the same time, and without stepping on each other's toes. It's a distributed version control system.

Our terminal prompt below is currently in a directory we decided to name "octobox". To initialize a Git repository here, type the following command:

git init



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

SEGURANÇA DO GIT

Medidas de segurança existentes no Git:

- *Commit signing*
  Chave GPG (GNUPg)

- *Push certificates*
  Man-in-the-middle

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

GARANTIAS DE SEGURANÇA

- Prevenir modificações dos dados comitados
- Garantir consistência do estado do repositório
- Garantir estado do repositório atualizado *(freshness)*

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

METODOLOGIA

I. **Identificação e descrição do ataque do metadado**

II. Projeto de um esquema de defesa para combater a manipulação dos metadados

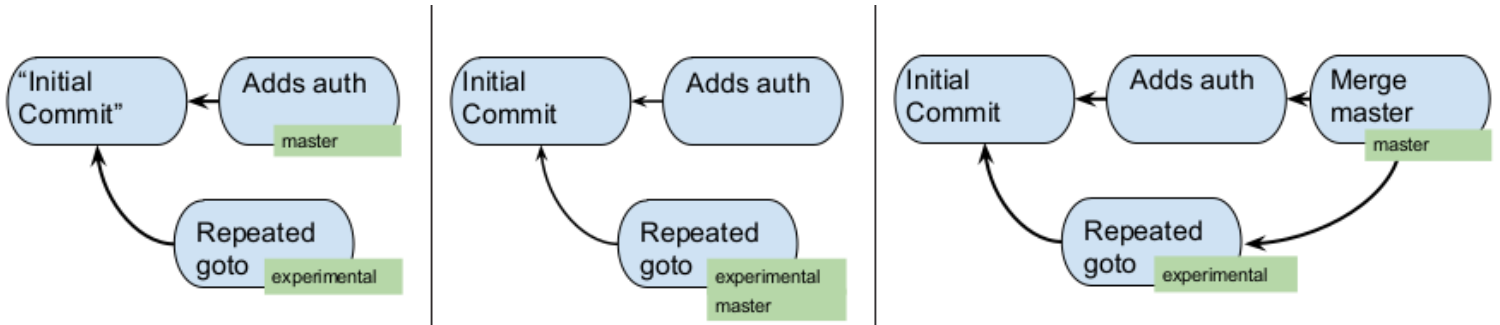
III. Implementar o esquema de defesa e estudar a sua eficiência

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

ATAQUES DE METADADOS

Três maneiras de alterar o branch:

- Acesso direto ao repositório
- Man-in-the-middle
- Manipulação do metadado localmente



(a) Original repository state (as cloned by the developer)

(b) The attacker changes the master branch pointer

(c) The developer pulled and, unknowingly, merged the experimental commit

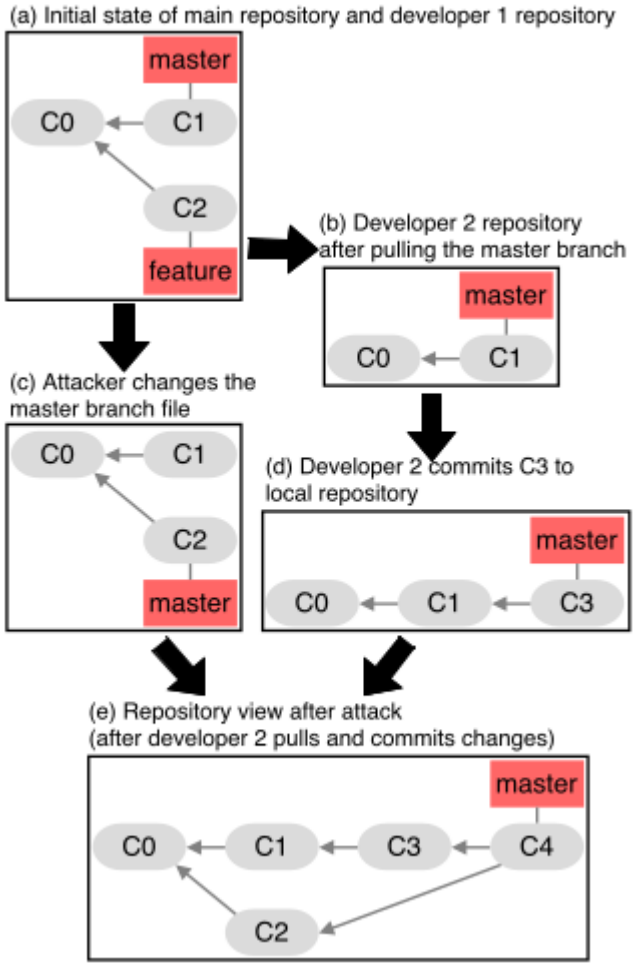
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

ATAQUES DE METADADOS

- *Teleport Attack*
  - *Branch*
  - *Tag*
- *Rollback Attack*
  - *Branch*
  - *Tag*
  - *Global*
- *Effort Duplication*
- *Deletion Attack*

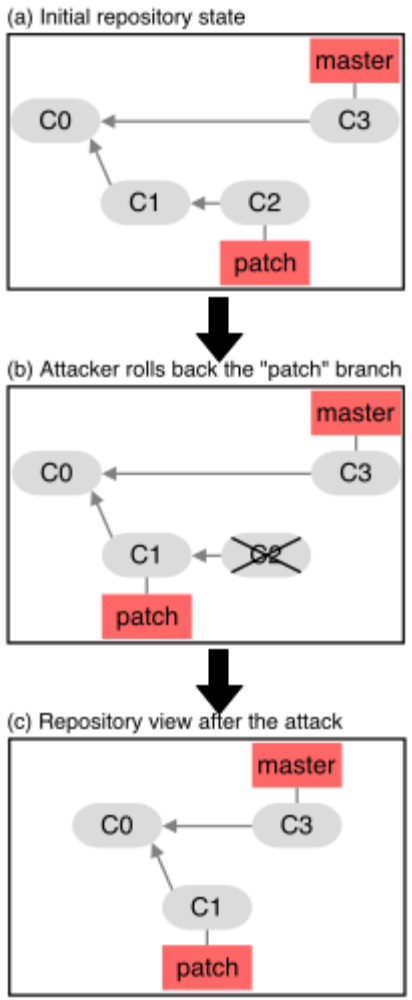
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

TELEPORT ATTACK

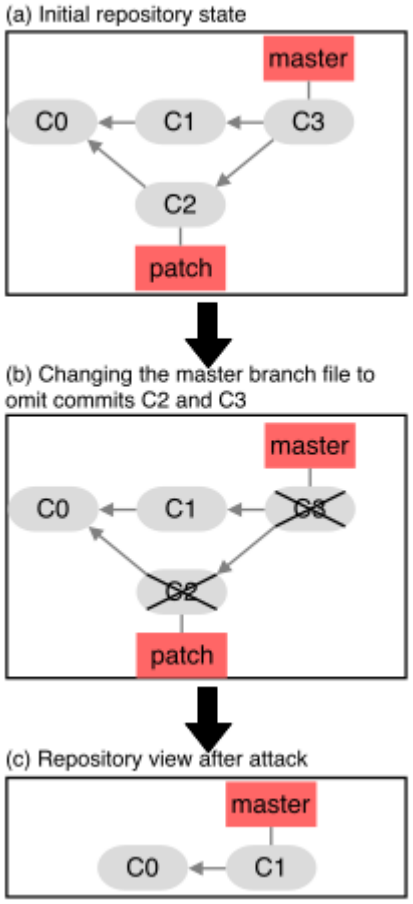


UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

ROLLBACK ATTACK



Branch rollback

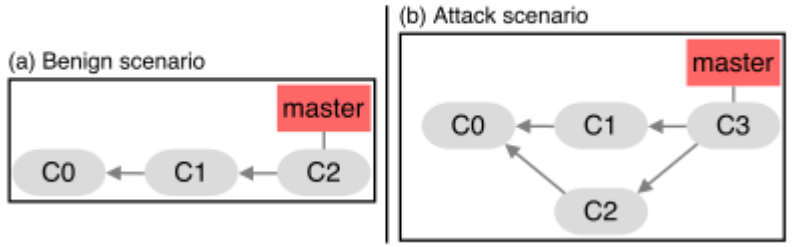


Global rollback

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

EFFORT DUPLICATION ATTACK

Variação do Global rollback



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

METODOLOGIA

I. Identificação e descrição do ataque do metadado

II. Projeto de um esquema de defesa para combater a manipulação dos metadados

III. Implementar o esquema de defesa e estudar a sua eficiência

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

OBJETIVOS

- Objetivo 1 (O1): Prevenir modificação dos dados comitados e garantir consistência do repositório.
- Objetivo 2 (O2): Preservar o fluxo de trabalho atual e ações comunmente usadas por desenvolvedores
- Objetivo 3 (O3): Manter compatibilidade com as implementações Git existentes

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

PROPOSTA

- Reference State Log (RSL)
- Nounce Bag

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

REFERENCE STATE LOG (RSL)

- Push Entry, para garantir que a localização da referência no momento do push

  Três possíveis retornos:

- *success*: Nenhuma ação necessária
- *fail*: Existe mudança no repositório remoto
  Precisa realizar *fetch* e *merge* localmente antes do push
- *invalid*: Validação RSL falhou (potencial ataque)
- Segue o mesmo fluxo de trabalho (*workflow*) do Git

PROCEDURE: Secure_push
Input: LocalRSL; related_commits; pushed reference X
Output: *result: (success/fail/invalid)*

```
1:  repeat
2:    result ← fail
3:    (RemoteRSL, nonce_bag) =
        Retrieve_RSL_and_nonce_bag_from_remote_repo
4:    if (RSL_Validate(RemoteRSL, nonce_bag) == false)
    then
5:      // Retrieved RemoteRSL is invalid
6:      // Must take necessary actions!
7:      return invalid
8:    if (new push entries for reference X in RemoteRSL) then
9:      // Remote repository contains changes
10:     // User must fetch changes and then retry
11:     return fail
12:   else
13:     prev_hash = hash_last_push_entry(RemoteRSL)
14:     new_RSL_Entry = create_push_Entry(prev_hash,
                                          nonce_bag, X)
15:     nonce_bag.clear()
16:     RemoteRSL.addEntry(new_RSL_entry)
17:     result = Store_in_remote_repo(RemoteRSL,
                                      nonce_bag)
18:     if (result == success) then
19:       // The remote RSL has no new entries
20:       push related_commits
21:       LocalRSL = RemoteRSL
22:       return success
23: until (result == success)
```

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

NONCE BAG

- *nonce*: número aleatório correspondente a um *fetch* do repositório principal
- *nonce bag*: lista não-ordenada de *nonces*
- Cada *nonce* do *nonce bag* serve como prova que foi apresentado um certo RSL ao usuário

PROCEDURE: Secure_fetch
Input: reference X to be fetched
Output: *result: (success/invalid)*

```
1:  repeat
2:    store_success ← false
3:    repeat
4:      (RemoteRSL, nonce_bag) =
          Retrieve_RSL_and_nonce_bag_from_remote_repo()
5:      fetch_success ← false
6:      // This is a regular "Git fetch" command.
7:      // Branch X's reference is copied to FETCH_HEAD
8:      fetch reference X
9:      C ← RemoteRSL.latestPush(X).refPointer
10:     if (C == FETCH_HEAD) then
11:       fetch_success ← true
12:   until (fetch_success == true)
13:   // Update the nonce bag
14:   if NONCE in nonce_bag then
15:     nonce_bag.remove(NONCE)
16:   save_random_nonce_locally(NONCE)
17:   nonce_bag.add(NONCE)
18:   // Storing the nonce bag at the remote repository
19:   //  might fail due to concurrency issues
20:   store_success = Store_in_remote_repo(nonce_bag)
21: until (store_success == true)
22: if (RSL_Validate(RemoteRSL, nonce_bag) == false)
    then
23:   // Retrieved RemoteRSL is invalid
24:   // Must take necessary actions!
25:   return invalid
26: else
27:   LocalRSL = RemoteRSL
28:   return success
```

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

RSL VALIDATE

- Utilizado para garantir que o RSL é válido
- Verifica se os push entries para dados RSL está corretamente ligado com outra

PROCEDURE: RSL_Validate
Input: LocalRSL (RSL in the local repository); RemoteRSL; nonce_bag
Output: `true` or `false`

```
1:  if (NONCE not in nonce_bag) and (NONCE not in RemoteRSL.push_after(LocalRSL) then
2:      return false
3:  // Verify that the ensuing entries are valid
4:  prev_hash = hash_last_push_entry(LocalRSL)
5:  for new_push_entry in RemoteRSL do
6:      if new_push_entry.prev_hash != prev_hash then
7:          // The RSL entries are not linked correctly
8:          return false
9:      prev_hash = hash(new_push_entry)
10: if verify_signature(RemoteRSL.latest_push) == false then
11:     // this RSL is not signed by a trusted developer
12:     return false
13: return true
```

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

ANÁLISE DE SEGURANÇA

- Prevenir modificações de dados comitados
  O mecanismo de assinatura do Git “dá conta” do recado

- Garantir consistência do estado do repositório
  O RSL provê uma visão consistente do repositório que é compartilhado com todos os desenvolvedores

- Garantir que o repositório sempre esteja atualizado (*freshness*):
  O *nonce bag* provê um repositório atualizado devido à aleatoriedade dos nonces

	Possible attacks	Time window of attack	Vulnerable commit objects
Commit signing	all attacks	Anytime	Any object
RSL (full adoption)	no attacks	None	No object
RSL (partial adoption)	all attacks	After the latest RSL entry and before the next RSL entry	Objects added after the latest RSL entry

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

ANÁLISE DE SEGURANÇA

Well behaving

Feature	Commit signing	Push certificate	RSL
Commit Tampering	✓	✓	✓
Branch Teleport	X	✓	✓
Branch Rollback	X	X	✓
Global Rollback	X	X	✓
Effort Duplication	X	X	✓
Tag Rollback	X	✓	✓
Minimum Git Version	1.7.9	2.2.0	1.7.9
Distribution Mechanism	in-band	(no default) or Additional server	in-band

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

METODOLOGIA

I. Identificação e descrição do ataque do metadado

II. Projeto de um esquema de defesa para combater a manipulação dos metadados

III. Implementar o esquema de defesa e estudar a sua eficiência

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

ANÁLISE DE SEGURANÇA

- Repositorios para avaliação

Repo.	Number of commits	Number of pushes	Repo. size	Repo. size with signed commits
R1	11,666	1,345	73.04	78.85
R2	7,521	26	66.96	69.79
R3	3,510	255	32.91	34.65
R4	6,031	194	15.79	19.98
R5	3,841	1,170	3.52	4.01

Overhead nos repositórios (storage)

Repo.	Our defense	Push certificates
R1	301.93	461.27
R2	6.49	8.88
R3	58.91	86.05
R4	44.34	66.27
R5	261.3	402.19

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

ANÁLISE DE SEGURANÇA

- Custo medio comunicação push

Scheme used	R1	R2	R3	R4	R5
Git w/ signed commits (baseline)	17.80	3,925.35	38.32	59.14	11.96
Our defense	44.01	3,950.87	63.56	84.71	37.65
Our defense (light)	28.28	3,935.71	48.61	69.52	22.28

Custo medio comunicacao por fetch

Scheme used	R1	R2	R3	R4	R5
Git w/ signed commits (baseline)	20.68	3,896.98	40.93	65.85	13.67
Our defense	46.18	3,922.40	66.48	91.27	38.77
Our defense (light)	35.19	3,911.81	55.84	80.67	28.01

- Atraso médio ponto-a-ponto por push

Scheme used	R1	R2	R3	R4	R5
Git w/ signed commits (baseline)	1.29	3.27	1.17	1.31	1.51
Our defense	3.11	5.27	2.78	2.95	3.51
Our defense (light)	2.44	4.49	2.16	2.40	2.81

Atraso médio ponto-a-ponto por fetch

Scheme used	R1	R2	R3	R4	R5
Git w/ signed commits (baseline)	0.87	1.95	0.75	0.66	0.67
Our defense	2.93	3.86	2.52	2.40	2.75
Our defense (light)	1.60	2.75	1.52	1.31	1.30

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

REFERÊNCIAS

[1] http://try.github.io

[2] http://git-csm.com

[3] https://www.wired.com/2010/03/source-code-hacks/

[3] http://www.geek.com/news/major-open-source-code-repository-hacked-for-months-says-fsf-551344/

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Perguntas?

Sugestões?

Obrigado!