DETECTING WEB ATTACKS WITH RECURRENT NEURAL NETWORKS

Arseny Reutov (raz0r@positive.com)
Fedor Sakharov (fedor_sakharov@sonm.com)

About us

Arseny Reutov (@theRaz0r) - application security researcher at Positive.com

Fedor Sakharov (@m0nt3kk1) -

software developer at sonm.com

Agenda

- The challenges of web attack detection
- Anomaly detection in HTTP requests with deep learning
- Demo, results & future work

THE CHALLENGES OF WEB ATTACK DETECTION

What Web Application Firewalls are

- Web Application Firewall (WAF) is a system that protects against application-level attacks (L7)
- First commercial WAFs appeared in 1999
- The most commonly known open-source WAF is mod_security (2002)
- Typically operate as a reverse proxy
- Most WAFs use pattern matching to detect attacks

Web attack types from WAF perspective

Time series-based:

- Web scraping
- Brute Forcing
- Fingerprinting
- Scanning
- L7 DDoS

HTTP Request/Response-based:

- SQL Injection
- Cross Site Scripting
- XML External Entities Injection
- Path Traversal
- OS Commanding
- Object Injection
- ...

Web attack types from WAF perspective

Time series-based:

- Web scraping
- Brute Forcing
- Fingerprinting
- Scanning
- L7 DDoS

HTTP Request/Response-based:

- SQL Injection
- Cross Site Scripting
- XML External Entities Injection
- Path Traversal
- OS Commanding
- Object Injection
- ...

We will focus on

Pattern matching

+ Effective to detect known attack vectors
+ Easily maintainable
+ Can be pretty fast
+ Predictable and interpretable behavior
+ Can work out of the box

- Subject to attacks, e.g. ReDoS
- Can be bypassed relatively easily
- Not so effective at catching unknown vectors aka 0-days
- Requires extensive web security domain knowledge
- Lots of false positives

Machine learning

+ Able to detect previously unseen samples
+ Usually not so easy to bypass
+ Once trained forward pass is pretty fast
+ Does not require web security domain knowledge

- Requires some time to be trained
- Results are difficult to interpret
- Unpredictable behavior
- Models are difficult to maintain

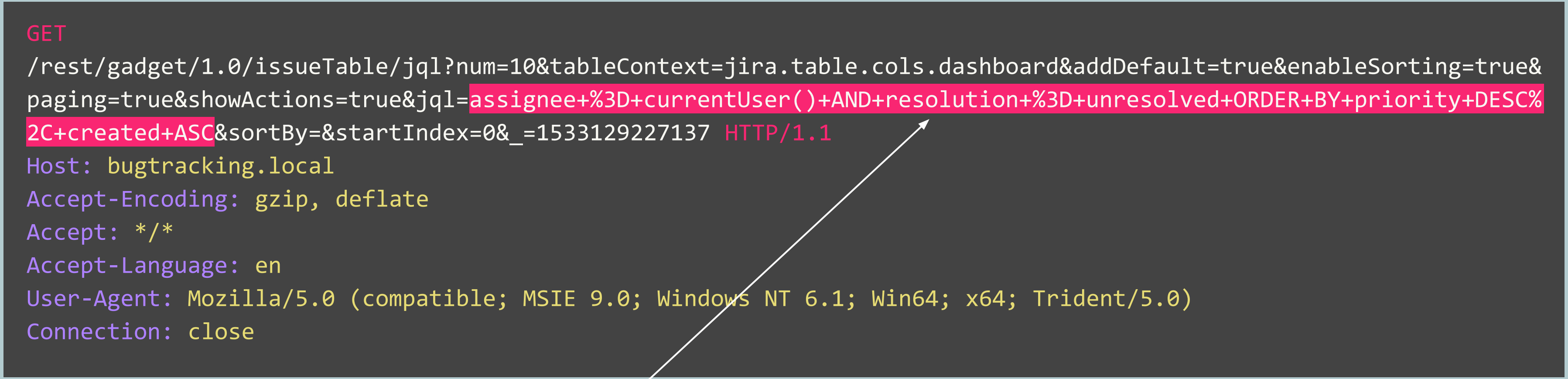
The goals of the research

- Create a deep learning model that does not require prior feature extraction
- The model should solve the task of anomaly detection in HTTP requests
- The model should yield interpretable results

What is an anomaly?

- Anomaly in an HTTP request can be anything: a request by curl, spam or even a 0day attack
- The model should understand the intention, whether it is negative (malicious) or not
- “Malicious/benign” classification greatly depends on context and history of previous observations

SQL Injection?



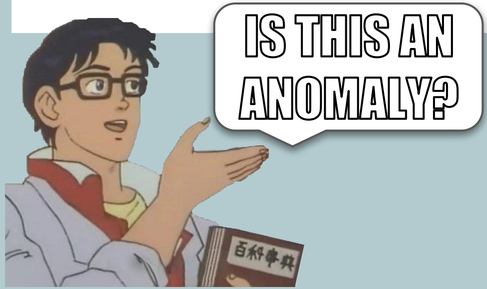
```
GET
/rest/gadget/1.0/issueTable/jql?num=10&tableContext=jira.table.cols.dashboard&addDefault=true&enableSorting=true&
paging=true&showActions=true&jql=assignee+%3D+currentUser()+AND+resolution+%3D+unresolved+ORDER+BY+priority+DESC%
2C+created+ASC&sortBy=&startIndex=0&_=1533129227137 HTTP/1.1
Host: bugtracking.local
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Connection: close
```

```
assignee = currentUser() AND resolution = unresolved ORDER BY priority DESC, created ASC
```

SQL Injection?

```
GET
/rest/gadget/1.0/issueTable/jql?num=10&tableContext=jira.table.cols.dashboard&addDefault=true&enableSorting=true&
paging=true&showActions=true&jql=assignee+%3D+currentUser()+AND+resolution+%3D+unresolved+ORDER+BY+priority+DESC%
2C+created+ASC&sortBy=&startIndex=0&_=1533129227137 HTTP/1.1
Host: bugtracking.local
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Connection: close
```

```
assignee = currentUser() AND resolution = unresolved ORDER BY priority DESC, created ASC
```



SQL Injection?

```
GET
/rest/gadget/1.0/issueTable/jql?num=10&tableContext=jira.table.cols.dashboard&addDefault=true&enableSorting=true&
paging=true&showActions=true&jql=assignee+%3D+currentUser()+AND+resolution+%3D+unresolved+ORDER+BY+priority+DESC%
2C+created+ASC&sortBy=&startIndex=0&_=1533129227137 HTTP/1.1
Host: bugtracking.local
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Connection: close
```

```
assignee = currentUser() AND resolution = unresolved ORDER BY priority DESC, created ASC
```

BENIGN

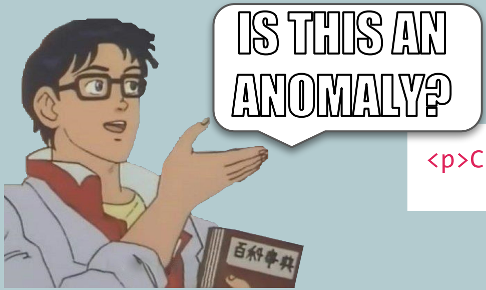
Cross Site Scripting?

```
POST /json/topic/?action=save HTTP/1.1
Host: habr.com
Connection: keep-alive
Content-Length: 129
Origin: https://habr.com
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139
Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Cookie: PHPSESSID=aasghtnlfls38i1f1n7hb5gn64;

id=&post_type=simple&title=&text=%3Cp%3ECheck+out+my+%3Ca+href%3D%22http%3A%2F%2Fhome.page%22%3Eblog%3C%2Fa%3E!%3
C%2Fp%3E&draft=1
```

```
<p>Check out my <a href="http://home.page">blog</a>!</p>
```

Cross Site Scripting?

```
POST /json/topic/?action=save HTTP/1.1
Host: habr.com
Connection: keep-alive
Content-Length: 129
Origin: https://habr.com
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139
Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Cookie: PHPSESSID=aasghtnlfls38i1f1n7hb5gn64;

id=&post_type=simple&title=&text=%3Cp%3ECheck+out+my+%3Ca+href%3D%22http%3A%2F%2Fhome.page%22%3Eblog%3C%2Fa%3E!%3
C%2Fp%3E&draft=1
```

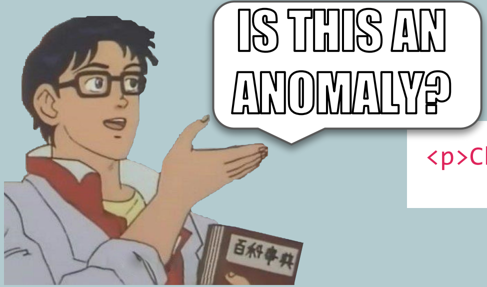


```
<p>Check out my <a href="http://home.page">blog</a>!</p>
```

Cross Site Scripting?

```
POST /json/topic/?action=save HTTP/1.1
Host: habr.com
Connection: keep-alive
Content-Length: 129
Origin: https://habr.com
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139
Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Cookie: PHPSESSID=aasghtnlfls38i1f1n7hb5gn64;

id=&post_type=simple&title=&text=%3Cp%3ECheck+out+my+%3Ca+href%3D%22http%3A%2F%2Fhome.page%22%3Eblog%3C%2Fa%3E!%3
C%2Fp%3E&draft=1
```

BENIGN



```
<p>Check out my <a href="http://home.page">blog</a>!</p>
```

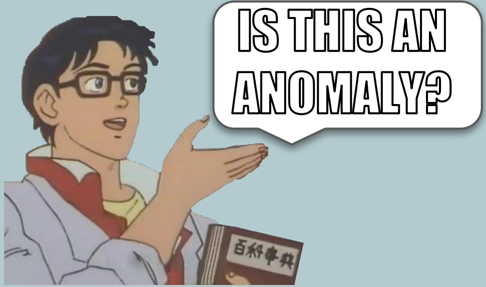
AI VILLAGE

Normal user registration?

```
POST /index.php/component/users/?task=user.register HTTP/1.1
Host: joomla.local
Connection: close
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Content-Length: 412
Content-Type: application/x-www-form-urlencoded

form[option]=com_users&user[password1]=password&user[username]=hacker&form[email2]=user@example.com&form[password2]=password&user[email2]=user@example.com&form[task]=user.register&user[password2]=password&user[name]=user&user[email1]=user@example.com&user[groups][]=7&form[name]=user&user[activation]=0&test=1&form[password1]=password&form[username]=user&form[email1]=user@example.com&user[block]=0
```

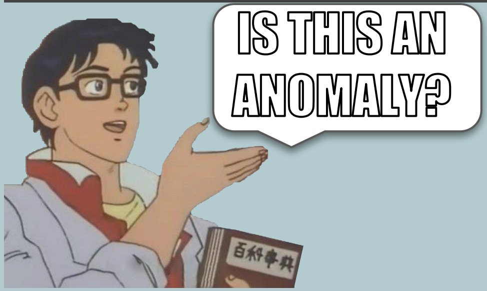
Normal user registration?

```
POST /index.php/component/users/?task=user.register HTTP/1.1
Host: joomla.local
Connection: close
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Content-Length: 412
Content-Type: application/x-www-form-urlencoded

form[option]=com_users&user[password1]=password&user[username]=hacker&form[email2]=user@example.com&form[password2]=password&user[email2]=user@example.com&form[task]=user.register&user[password2]=password&user[name]=user&user[email1]=user@example.com&user[groups][]=7&form[name]=user&user[activation]=0&test=1&form[password1]=password&form[username]=user&form[email1]=user@example.com&user[block]=0
```



Normal user registration?

```
POST /index.php/component/users/?task=user.register HTTP/1.1
Host: joomla.local
Connection: close
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Content-Length: 412
Content-Type: application/x-www-form-urlencoded

form[option]=com_users&user[password1]=password&user[username]=hacker&form[email2]=user@e      le.com&form[password
2]=password&user[email2]=user@example.com&form[task]=user.register&user[password2]=passw      r[name]=user&user[
email1]=user@example.com&user[groups][]=7&form[name]=user&user[activation]=0&test=1&fo      d1]=password&form
[username]=user&form[email1]=user@example.com&user[block]=0
```



Joomla <3.6.4 Privilege Elevation

ANOMALY DETECTION IN HTTP REQUESTS WITH DEEP LEARNING

Take one: try to build a classifier

- Collect some benign data
- Generate some malicious data
- Try to build a classifier:

Take one: try to build a classifier

- Collect some benign data
- Generate some malicious data
- Try to build a classifier:

Sample	Label
GET /api/posts?author=mallory&category='%20or%20'1'%20=%20'	1
GET /api/posts?author=alice&category=sports	0

Take one: try to build a classifier

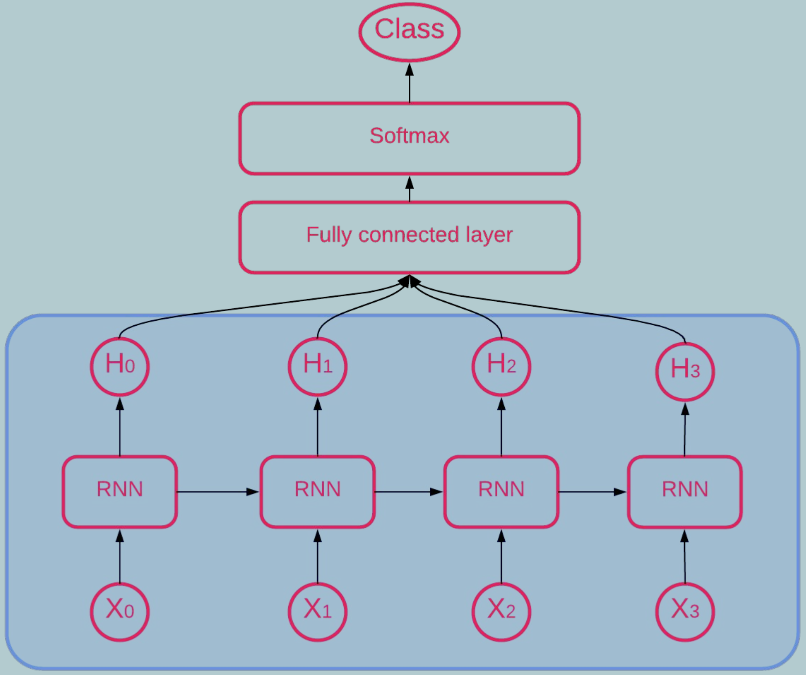
- HTTP is a text-based protocol
- Each line is an independent sentence
- Headers, URI are not that long
- Sequential nature, e.g. the value of parameter depends on its name

```
POST /vulnbank/online/api.php HTTP/1.1
Host: 10.0.212.25
Connection: keep-alive
Content-Length: 59
Accept: application/json, text/javascript,
*/*; q=0.01
Origin: http://10.0.212.25
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (X11; Linux x86_64)
AppleWeb...
```

```
Now is the winter of our discontent
Made glorious summer by this sun of York;
And all the clouds that lour'd upon our house
In the deep bosom of the ocean buried.
Now are our brows bound with victorious wreaths;
Our bruised arms hung up for monuments;
Our stern alarums changed to merry meetings,
Our dreadful marches to delightful measures.
```

Take one: try to build a classifier

- RNNs are used for analyzing sequential data
- Build a classifier
- Evaluate results
- Somewhat good, however...



RNN (unfolded)

Take one: try to build a classifier

- RNNs are used for analyzing sequential data
- Build a classifier
- Evaluate results
- Somewhat good, however...

There are problems:

- Results are not interpretable (we only get a label)
- Construction of malicious classes is tricky
- Needs manual labeling

Take two: try to improve classifier

- Add attention layer
- Attention aids learning process
- And helps interpreting model’s decisions
- But it doesn’t solve other problems with classification

Take three: anomaly detection

- What about anomaly detection?
- The initial task of attack is more similar to it
- No longer have to manually label data
- And no need to generate malicious samples

Take three: anomaly detection

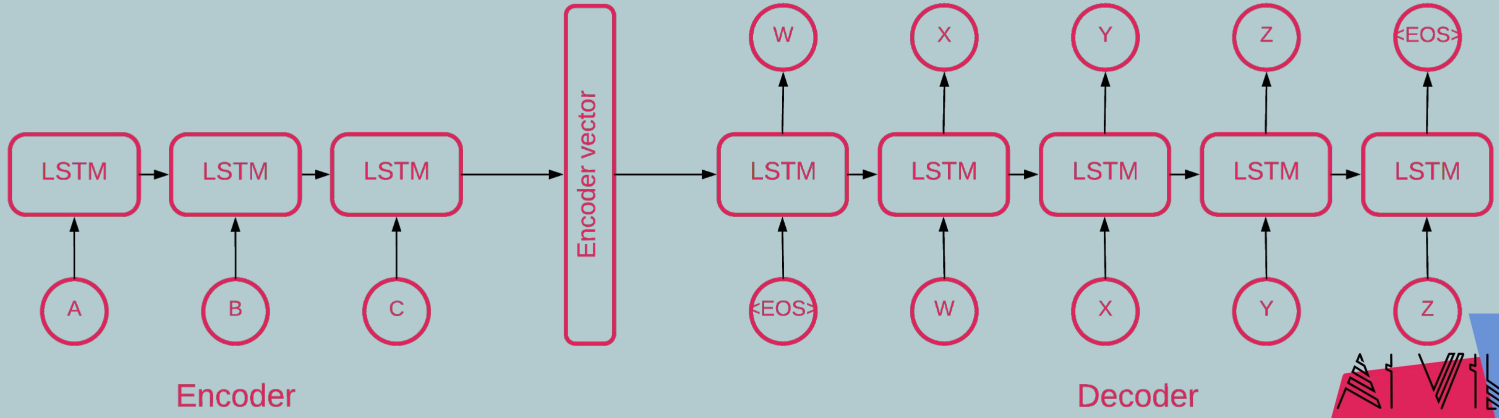
Let’s take a look at this model for machine translation:

- Uses two multi-layered LSTMs: encoder and decoder
- Encoder maps input to vector of fixed dimensionality
- Decoder decodes the target vector using this vector

Take three: anomaly detection

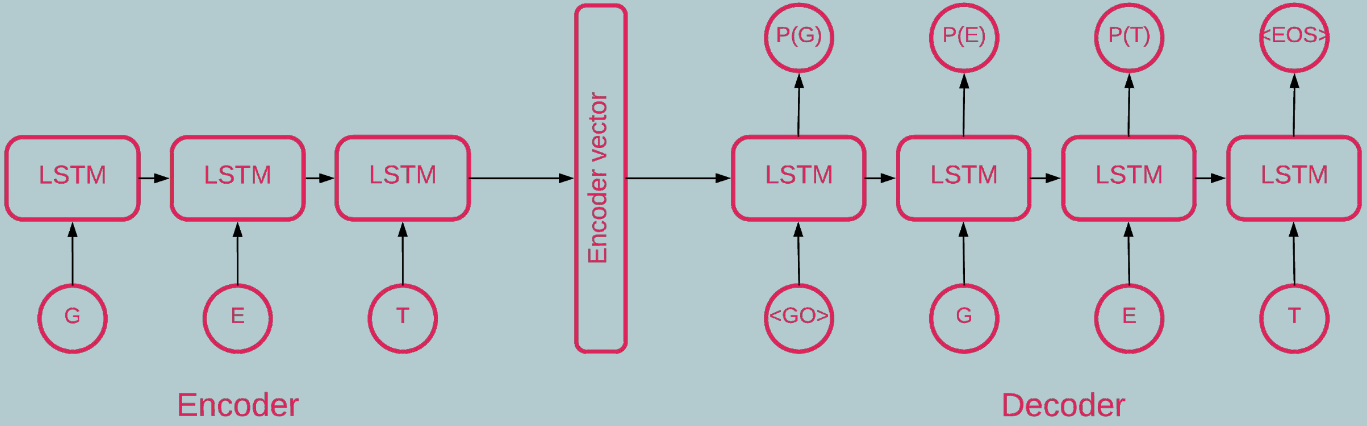
Let's take a look at this model for machine translation:

- Uses two multi-layered LSTMs: encoder and decoder
- Encoder maps input to vector of fixed dimensionality
- Decoder decodes the target vector using this vector



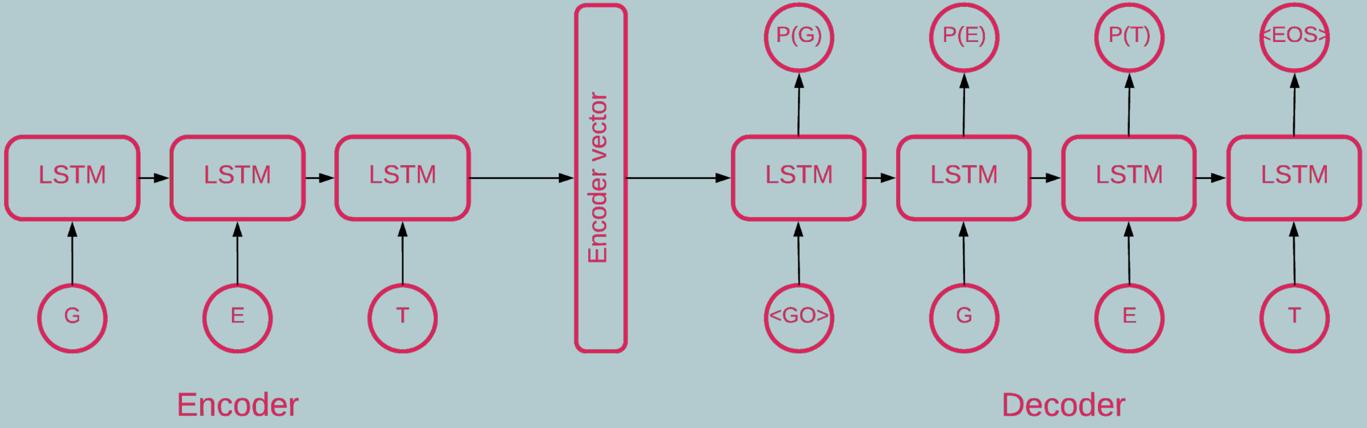
Take three: anomaly detection

But if we feed inputs also as target outputs the model will learn to reconstruct the sequences that it has seen:



Take three: anomaly detection

- Now the model outputs the probabilities of each letter in the sequence and also the loss:

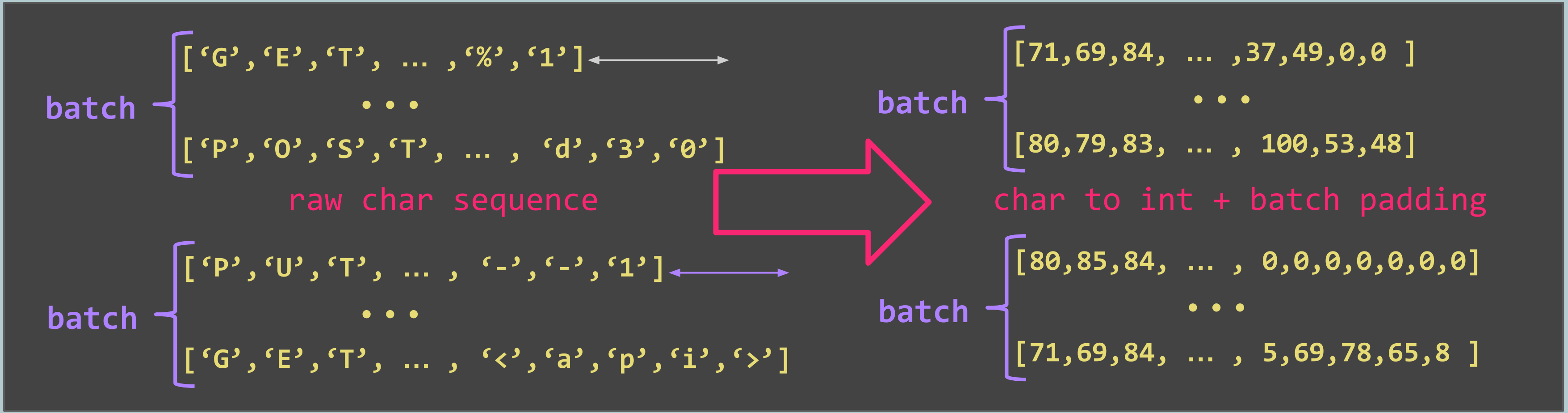


Take three: anomaly detection

- Now the model outputs the probabilities of each letter in the sequence and also the loss
- All requests with a “high” loss are considered as malicious
- For these requests probabilities for “anomalous” characters are low

Take three: anomaly detection

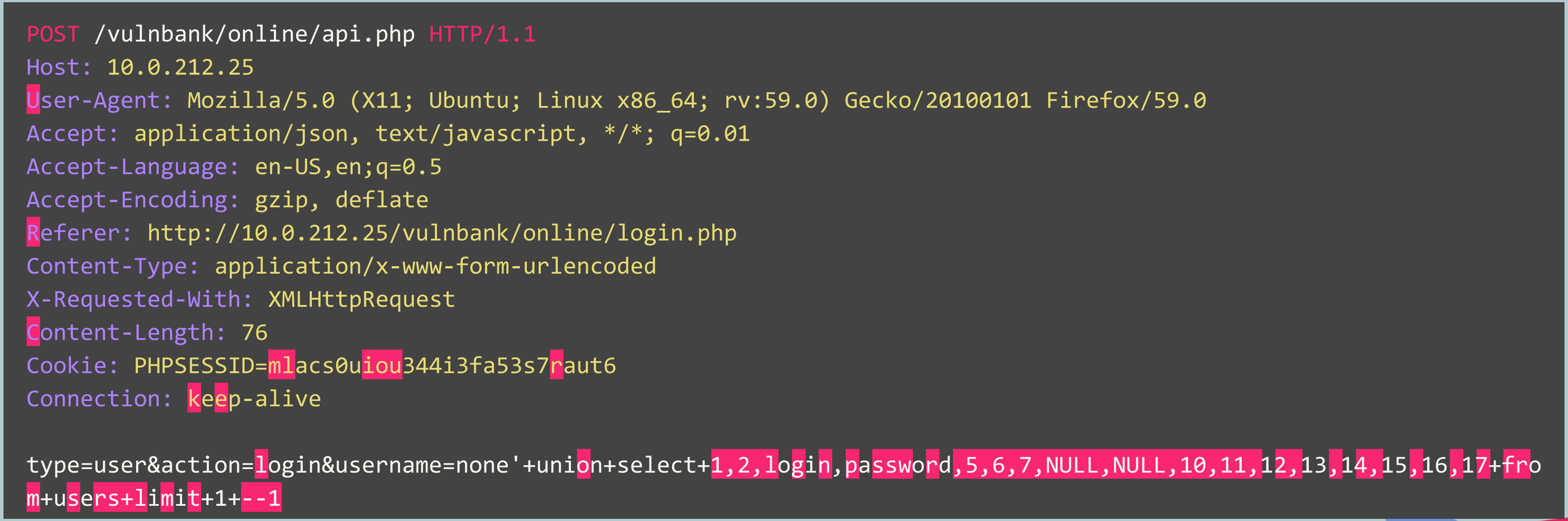
The input is transformed from strings with different length to integers using a dictionary (vocab.json) and padded to max length in the batch.



```
      [‘G’,‘E’,‘T’, … ,‘%’,‘1’]                      [71,69,84, … ,37,49,0,0 ]
batch          • • •                           batch           • • •
      [‘P’,‘O’,‘S’,‘T’, … , ‘d’,‘3’,‘0’]             [80,79,83, … , 100,53,48]
             raw char sequence                       char to int + batch padding

      [‘P’,‘U’,‘T’, … , ‘-’,‘-’,‘1’]                 [80,85,84, … , 0,0,0,0,0,0,0]
batch          • • •                           batch           • • •
      [‘G’,‘E’,‘T’, … , ‘<’,‘a’,‘p’,‘i’,‘>’]         [71,69,84, … , 5,69,78,65,8 ]
```

Take three: anomaly detection

If the anomalous request was to be visualised:

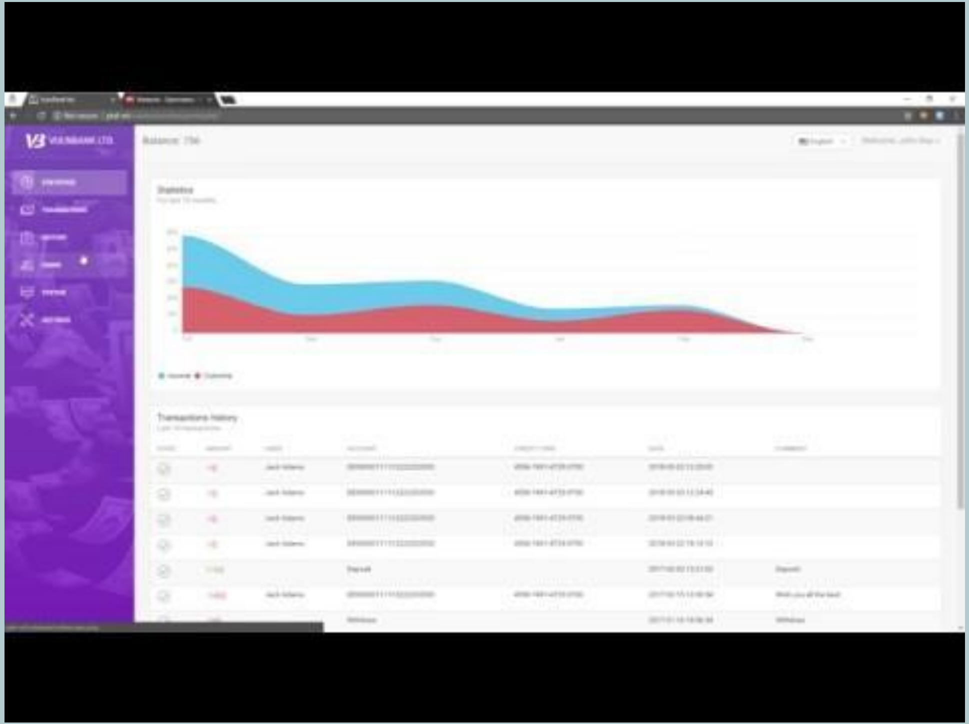


```
POST /vulnbank/online/api.php HTTP/1.1
Host: 10.0.212.25
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:59.0) Gecko/20100101 Firefox/59.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.0.212.25/vulnbank/online/login.php
Content-Type: application/x-www-form-urlencoded
X-Requested-With: XMLHttpRequest
Content-Length: 76
Cookie: PHPSESSID=mlacs0uiou344i3fa53s7raut6
Connection: keep-alive

type=user&action=login&username=none'+union+select+1,2,login,password,5,6,7,NULL,NULL,10,11,12,13,14,15,16,17+fro
m+users+limit+1+--1
```

DEMO, RESULTS & FUTURE WORK

AI VILLAGE

It’s Showtime!



VULNBANK LTD

Balance: 756

Statistics

Transactions History

The goals and results of the research

- Create a deep learning model that does not require prior feature extraction ✓
- The model should solve the task of anomaly detection in HTTP requests ✓
- The model should yield interpretable results ✓

https://github.com/PositiveTechnologies/seq2seq-web-attack-detection

Future work

- Optimize learning time (now takes ~5 hours on a GPU for a 300 Mb dataset)
- Build one more model on top of it to classify the anomalous sequences
- Improve threshold calculation

THANK YOU

QUESTIONS?