An Introduction to Printer Exploitation

Exploit Development tutorial, iot, exploitation, hardware

ricksanchez Leader 2 Sep '17

> Preface

Note: As always the following is just a digest of all the things I could observe by working on printers myself or facts from stuff I read about recently.

Since this thread about the HP printer promo videos caught some attention I will try to shed some light onto the field which was displayed there.
First of all we should keep in mind this was a *promo video* made by a company.
So always ask yourself this: “How real are the displayed scenarios, or are these just ‘Hollywood fabrications’?”

I had some access to different printers over the last couple of month and learned some basic principles, which I wanna share with
you as good as possible now.
Printer use a various amount of protocols and firmwares which differ from vendor to vendor and model to model.
So this first part might be boring to some, you can try to skip the theoretical part and jump right to the exploitation paragraph, but talking about fundamentals will cover important topics.

> Printer as an attack vector?

- So why would I even want to target a printer in the first place?
- Why not just target Desktop or Server environments with malware as usual?

We get to that in next couple of paragraphs

Author Assigned Level: Newbie

Community Assigned Level:

◯ Newbie ◯ Wannabe ◯ Hacker ◯ Wizard ◯ Guru	61 voters
Show results	

Required Skills

Not much to mention here

- basic ability to read for more than 5 minutes

> Printer a viable target or just wasted time?

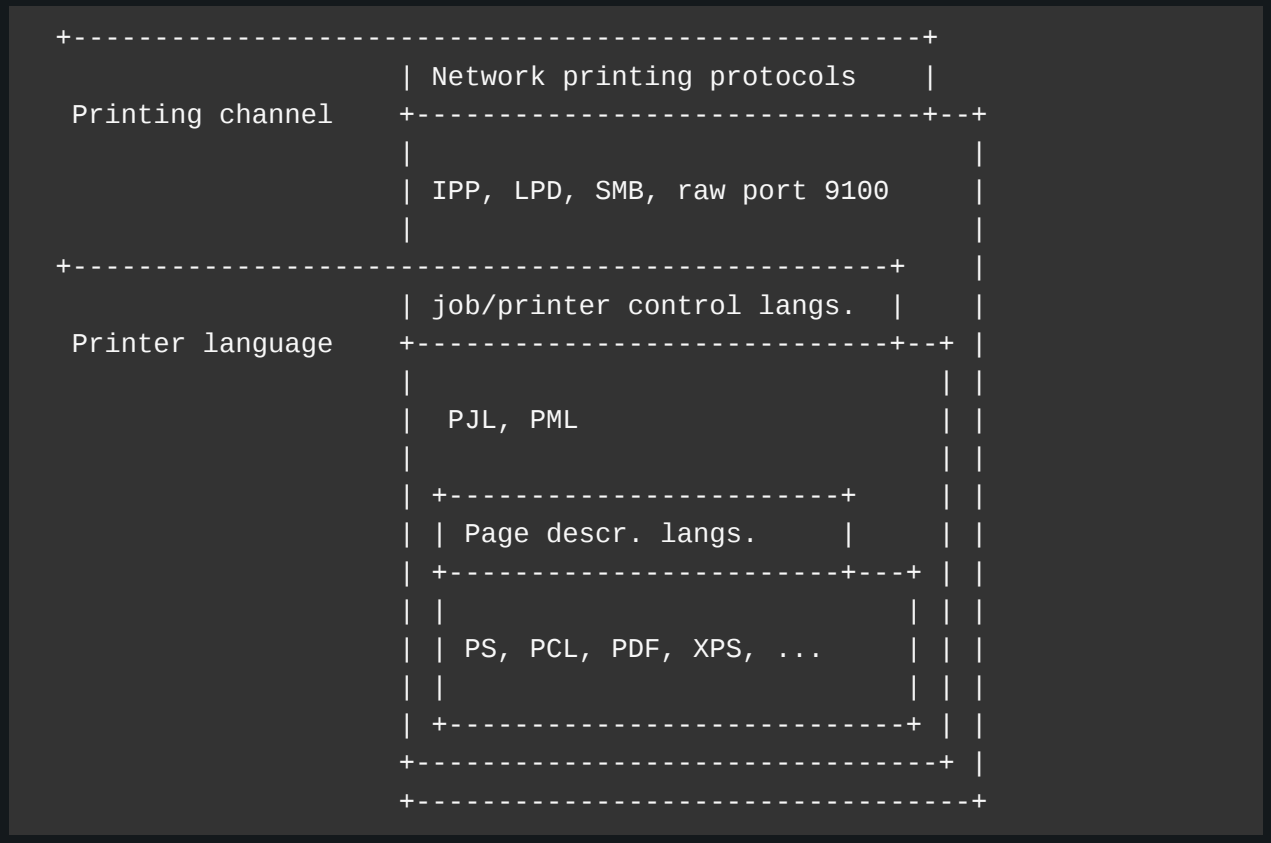
> Local vs Network printers

Local printers are just directly connected to a desktop PC and are rather uninteresting.
These days almost all printers seem to be network printers though.
So basically network printing enables users in locations geographically separate from each other and from their print devices to produce documents for themselves and others.
Print servers enable multiple clients to share one or more print devices.

So far so easy right?
Let's jump directly to some highlevel view which explains every network printer quite well.

> Highlevel view

A highlevel view of current network printers might look something like this:



Note: This diagram might be incomplete!

=> The network printing protocol acts as a channel to deploy print jobs, which either contain the page description language directly or first invoke a printer/job control language!

Let's take a look at each of those sections in the diagram above more closely and cover some fundamentals.

> Fundamentals

> Firmware

Printer use, in my experience a couple of different operating systems for embedded devices.
I'll list a few of them here, but won't really dive into them, since it would go beyond the scope of this article.

- Basic but slimmed down GNU/Linux, often custom tailored,
- WindRiver Linux 44,
- VxWorks 30,
- ThreadX 24.

With the different, but limited pool of printers I've had access to all of them had some things in common in the end.

- slimmed down instruction/command set - reduced functionality,
- 'legacy kernels' - often around kernel version 2.6.XYZ,
- might include 'hidden' functionality, which can be enabled through a little patch 😉 - e.g.: ssh files are there, but need to be enabled in config files,
- ssh is more present in printers designed for offices, compared to home printers for some reason,
- sometimes the way the firmware is stored is hilarious - e.g.: on a SD card you can remove/switch within 30 seconds of physical access

These facts show that printers might be vulnerable to certain attacks, but still these attacks often are made more 'complicated', because certain functions aren't even there or somehow have to get enabled through (remote) file system writes…

Next a wild bunch of protocols is used for communication between Printers, print servers, desktop PCs and even internally within a printer.
Let's take a look!

> Network printing protocols

To summarize it right away there are a bunch of ‘exotic’ protocols for network printing (NCP or AppleTalk for example)
To explain and mention them all here would be too much again.
If anyone is interested in some specifics or a follow up post I’d answer any questions there.

In the Windows world, SMB/CIFS printer are popular.
The most common printing protocols supported directly by network printers however are LPD, IPP, and raw port 9100 printing, which I will explain a bit more in depth now.
Furthermore, some devices support printing over generic protocols such as FTP or HTTP file uploads as well.

> LPD

LPD is short for ‘Line Printer Daemon’-protocol.
It runs on port 515/TCP and can be accessed by using ‘lpr’ over the CLI.
To print things, the client sends a control file defining job/username and a data file containing the actual data to be printed.

> IPP

IPP is an extendable protocol and based on HTTP, so it inherits all existing security features like basic authentication and SSL/TLS encryption.
To submit a print job, a HTTP POST request is sent to the IPP server, which listens on 631/TCP.
For anyone wondering CUPS is an IPP implementation, which is a default printing system in many Linux distributions and macOS X.

> SMB

SMB, short for ‘Server Message Block’ is an application-layer network protocol, which handles file and printer sharing.
It’s used by default on Windows.
Usually it runs on 445/TCP.

> Port 9100

Also known as ‘raw printing’, since it makes use of connecting to 9100/TCP of a network printer.
It is the default method used by CUPS and the Windows printing architecture.
Here all data sent is directly processed by the printing device, just like a parallel connection over TCP.
In contrast to LPD, IPP and SMB interpreted printer control/page description languages, this one here is capable of sending direct feedback to the client, including status and error messages.
So we have a *bidirectional channel* here, which directly can give us access to results of the Printer control languages!

> Printer Control Languages

Basically a job control language manages settings like output trays for the current job.
It often just sits in between the printing protocol and the page description language.
Printer control and management languages are designed to affect not only a single print job but the device as a whole.
I’m not too knowledgeable here but the two most basic ones are listed below.

> SNMP

SNMP, short for ‘Simple Network Management Protocol’ listens on 161/UDP.
Was designed to manage network components

> PJL

PJL, short for ‘Printer Job Language’ is the kinda de-facto standard now
Can be used to manipulate general settings, also with permanent changes.
There are many dialects as vendors tend to support only a subset of the commands listed in the PJL reference and instead prefer to add proprietary ones.
PJL is also used to set the file format of the actual print data to follow, which makes it interesting for various attacks.

> Page Description Languages (PDL)

This one basically specifies how the actual document will look like appearance wise.
Here comes the printer driver into play which kinda translate the file to be printed into a PDL that is understood by the printer.

> PostScript (PS)

Is well known and made by Adobe and is widely used as a PDL.
PS is capable of far more than just defining the appearance of the document and handling vector graphics though.
That's why, when used correctly, PS can be used for a variety of attacks such as denial of service (for example, through infinite loops), print job manipulation and retention as well as gaining access to the printer's file system.

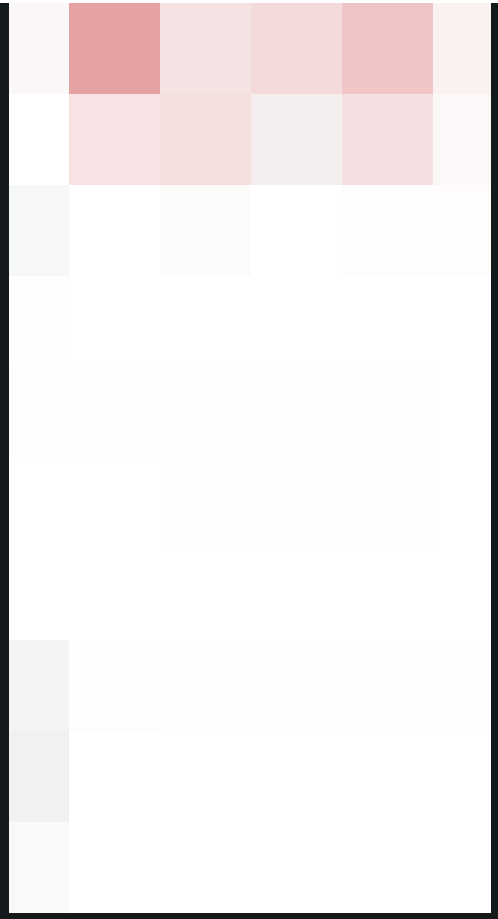
> PCL

As a minimalist page description language supported by a wide variety of vendors and devices.
Is also a de-facto Standard nowadays.
It's also not intended to get direct access to the underlying filesystem.
So it's not that well suited for exploitation purposes, but still has it's place for such purposes as well.

> Possible Exploits

> Who would put a printer on the Internet?

I just leave this Picture as a first expression here 😉



Data from: 07.Sep.2017

> Attack Vectors

> Remote

As easily seen above a lot of printers are connected to the *Internet* through port 9100, which make them attackable.
You either know the IP or can just scan for some in your neighborhood radius/ check shodan.

Once you have some you might get a SSH connection going.
Often standard login credentials are still used, which you can easily scrape from the Internet…

> Inside job

If you have physical access to the printer you can also plug in an USB drive or even a SD card.

> Possible Mayhem one can cause…

So now we're kinda back to the linked topic at the beginning of the small web series directed by HP.
So how realistic are the shown scenarios?

> DoS

- Transmission Channel - basically block the/one printing port to keep the printer busy and don't print anything anymore.
- Document processing - manipulate a Document via PDL and let the printer interpret it… e.g.: an infinite loop in PS.
- Physical damage - malware causing writes on NVRAM chips which have a life expectancy of ~10^5 writes [25].

> Privilege Escalation

- Factory defaults - reset to factory defaults to bypass authentication.
- Accounting bypass - similar thing here, printing without authentication.

> Print Job Access

- Print job retention - Try to find stored print jobs on the printer and extract those.
- Print job manipulation - Alter print jobs. You can imagine the possible mayhem caused itself.

> Information Disclosure

- Memory access - may lead to finding sensitive data like passwords or printed documents.
- File system access - potentially retrieve sensitive information like configuration files or stored print jobs.

- Credential disclosure - brute force attacks against changed default login credentials to gain access

> Code Execution

- Buffer overflows - printers provide additional languages and network services, potentially prone to this kind of attack
- Firmware updates - it is common for printers to deploy firmware updates as ordinary print jobs 😉 **cough** malicious firmware **cough**
- Software packages - ‘custom tailored and manipulated printer apps’

> Misc

- Malware - target network printers and spread it in local networks to other peers.

> possible scenarios

Depending on the planned attack and possible access one has a variety of attack vectors.
One need more planning than others.
Some need physical access and some can be done from remote.
Combinations of those are easily possible!

For example issuing a malicious firmware update via a simple print job (possible case: no authentication needed), which extracts sensitive data and renders the printer useless.
-> Printer ‘ransomware’ may be a thing, even if it sounds kinda weird.

So to conclude this section, I think the shown attacks in the videos were presented a tad to ‘flashy’, but are indeed possible depending on the printers and network they are placed in.

> Tools

A lot of these techniques mentioned above need some serious work or knowledge about the underlying structure (e.g.: used PDL, PCL).
Even though these might be fairly easily found out using manuals or online search it’s still a hassle and

extra work.
So people already made our lifes more easy by providing tools for almost all tasks mentioned above :).

> BeEF

The Browser Exploitation Framework (BeEF) is a penetration testing tool that focuses on the web browser.
It allows the penetration tester to assess the actual security posture of a target environment by using client-side attack vectors.
This is not really printer specific, **but** it is a framework to implement cross-site printing 38 functionality.

> Praeda

Praeda - “An Automated Printer Data Harvesting Tool” written in perl.
Also a tool to help pentesters to gather usable data during security assessment jobs.
Praeda systematically collects sensitive information from the printer’s embedded web server.
This includes device passwords, usernames, email addresses which might be available publicly on the web interface.

> PRET 221

This one is real nifty tool written in python to check for basically every attack vector I mentioned above.
It tries to connect to the printer via network or USB and tries to exploit the used printer languages, currently supported are PS, PJL and PCL.
When successfully connected one has a ton of available commands.
A full list can be found on the Github, linked below.

> LES

Linux Exploit Suggester is a neat little perl script, which gives some options for possible exploits depending on your kernel.
As stated above the kernel versions for embedded operating systems are often far lower, compared to current linux based desktop or server distributions.
So old, usually fixed exploit techniques might still be viable here!

Note: It is likely, that perl is not present in it's full range and copying it to a printer is extra work.
Luckily one can run simply run in a desktop environment and specifying the kernel you want to exploit

---

> My home printer - a journey to find a way in!

Ok what is a basic plan to concentrate on when trying to exploit a printer?
I've given a lot of theory until this point, as well as some "Do's" and "Mights".
Maybe you've got some ideas on your own already, but here's a little experimental journey from me.

So first thing that is obvious is to check for open ports and an OS fingerprint.
Luckily we have nmap.
Nmap is bae for this.

> Where's the door?

```
$ sudo nmap 192.168.1.108
Starting Nmap 7.01 ( https://nmap.org ) at 2017-09-11 20:13 CEST
Nmap scan report for 192.168.1.108
Host is up (0.031s latency).
Not shown: 993 closed ports
PORT     STATE SERVICE
80/tcp   open  http
139/tcp  open  netbios-ssn
443/tcp  open  https
445/tcp  open  microsoft-ds
515/tcp  open  printer
631/tcp  open  ipp
9100/tcp open  jetdirect
MAC Address: 44:D2:44:1C:73:E2 (Seiko Epson)

Nmap done: 1 IP address (1 host up) scanned in 2.04 seconds

Device type: specialized
Running: Linux 2.6.X
```

```
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.31 - 2.6.35 (embedded)
Network Distance: 1 hop
```

So we have the usual printing ports open, as well as some other basic ones.
It is running an older Linux as well, so no big surprise there!
No open 22/TCP port though.
So causing mayhem on the file system is not possible as of now.

> PRET and done?

I've praised PRET quite a bit above, so let's give it a try to check if my Epson printer has a nice, hopefully standard set of supported printer languages!

```
$ python pret.py 192.168.1.108 -s PS

Checking for IPP support:       found
Checking for HTTP support:      found
Checking for SNMP support:      found
Checking for PS support:        not found
$

$ python pret.py 192.168.1.108 -s Pjl
Checking for IPP support:       found
Checking for HTTP support:      found
Checking for SNMP support:      found
Checking for PJL support:       not found
$

$ python pret.py 192.168.1.108 -s PCL
Checking for IPP support:       found
Checking for HTTP support:      found
Checking for SNMP support:      found
```

```
Checking for PCL support:       not found
$
```

So no SSH and not even a standard version here…
Most likely the result of my vendor using some exotic stuff again and not keeping things simple 😕 …

Anyway using PRET is easy and self explanatory, once connected a help function will give you an overview of available stuff!
From checking the file-system. creating directories, changing configuration files or even dumping the whole NVRAM.
PRET can do it all (in theory that is 😉).

After trying a few things to find a way to make PRET work for me I trashed that idea for now and moved on!

> LES

So I wanted to have some fun now after the two disappointing results 😃 :D.
So let's dig deeper into what Linux exploits might get suggested for our version!

```
$ perl Linux_Exploit_Suggester.pl -k 2.6.31

Kernel local: 2.6.31

Searching among 65 exploits...

Possible Exploits:
[+] american-sign-language
   CVE-2010-4347
   Source: http://www.securityfocus.com/bid/45408/
[+] can_bcm
   CVE-2010-2959
   Source: http://www.exploit-db.com/exploits/14814/
[+] do_pages_move
   Alt: sieve    CVE-2010-0415
   Source: Spenders Enlightenment
```

```
[+] half_nelson
   Alt: econet    CVE-2010-3848
   Source: http://www.exploit-db.com/exploits/6851
[+] half_nelson1
   Alt: econet    CVE-2010-3848
   Source: http://www.exploit-db.com/exploits/17787/
[+] half_nelson2
```

Note: If these are viable and meet all dependencies has to be checked of course, but a brief look at them made me decide not to spend too much effort here.

> Manual PJL Injection

So I thought why not check for PJL again and try invoking some command strings manually in combination with netcat as a listener!

So I tried using:

```
echo "@PJL FSUPLOAD FORMAT:BINARY NAME="../../etc/passwd" OFFSET=0 SIZE=648"
# If successful this should display the */etc/passwd* file.
```

or

```
echo "@PJL INFO ID" | nc -v -v 192.168.1.108 9100
# If successful this should get the *printer's device information*
```

as well as other PJL command injecting techniques, but my printer is not accepting any of these.
It's not reacting at all to this kind of 'attack'…

I'm not knowledgeable enough to launch this with PS and PCL as well, because their command syntax differs greatly (obviously).
I'm remaining with a note to search for PS and PCL attack strings.

> A PRET test script to the rescue?

So PRET doesn't work for my home printer as seen above.
Interestingly I found that there is a script "hidden" within the PRET source folder called "lpdtest.py"
It can test for known, but older (like really older) vulnerabilities within the Line Printer Daemon, listed here [41].
This involves some basic tests:

'get' Test

Trying to get (aka print) a file from printer's file system.

$ lpdtest.py printer get /etc/passwd
$ lpdtest.py printer get …/…/…/etc/passwd

etc…

'in' Test

This test is for fuzzing around with user input (hostname,username, jobname, filenames, etc.).
This might be useful to test for interpretation of shell commands…

```
# Test for environment variables
$ lpdtest.py printer in '$UID'

# Test for pipes and redirects
$ lpdtest.py printer in '| pwd'
$ lpdtest.py printer in '>> /etc/passwd'

# Test for backticks
$ lpdtest.py printer in '`ls`'

# Test for [shellshock (CVE-2014-6271)](http://seclists.org/oss-sec/2014/q3/
$ lpdtest.py printer in '() {:;}; /bin/ping -c1 1.2.3.4'
```

As expected these attacks were already fixed.
My printer spit out a few pages with lines like

“If you can read this lpdtest.py XYZ failed!”

So the result here some wasted paper and ink…

Summary

Why Printer Exploitation?

- (most) printers are already full blown computers!
  - Printer as port/network/exploits scanner
  - Computing/hash-cracking/sniffing
  - Malware upload
  - “Stealth”/“uncleanable” command and control
  - Unencrypted data theft

Afterthoughts

- How many people would expect their printers is infected?
- How many users/admins/security-auditors audit and hard secure their network printers?
- How many persons or anti malware products could clean such a malware?
- …?

Outlook and closing words

If I get the hands on some nicer printer I will deliver some exploit stuff later on I promise.
If I get some more time to get a breakdown of my current home printer so I can take a look under the hood and to figure something out.
An example here would be to capture a firmware update and trying to unpack/reverse that one.
This would take a lot more time and preparation of my part, which would cause serious delay to this article as well.

So I'm keeping it rather open ended now, but I hope I could inspire some minds here to take a closer look as well.
Furthermore I hope this article reached the people who were interested and were able learn some things.
So if you want to try to exploit your own device, just try it out!
Remember:

- Find a way into the system,
- Check for the used printer languages and try code injection techniques for these,
- Try dumping the file system directory structure from the web interface,
- Upload self created "malicious" firmware if it is supported,
- Find a new way 😉

I'm looking forward to feedback and improvement suggestions.

Further readings

Article related resources:

- LPD RFC 9
- SMB RFC 10
- IPP RFC 13
- How Network Printing Works 9
- PostScript Manual 2
- BeEF 14
- Praeda 7
- PRET 221
- Linux Exploit Suggester 23
- Printer Security Test Cheat Sheet 28
- Hacking Printers Wiki 18

Extras:

- Running DOOM on a Printer 45
- From patched to Pwnd 18
- Thousands of printers hacked across the globe after critical flaw exposed 21

- Cross_Site_Printing 18

24

[Beta] The New Bounty Program 6

created	last reply	replies	views	users	likes	links
Sep '17	Dec '17	13	19.8k	9	41	23

Sirius First 100 Sep '17

The amount of network enabled printers I find during engagements is astounding.

Excellent write-up, will definitely be coming back to this one for some lulz.

5

ricksanchez Leader Sep '17

> Sirius:
>
> The amount of network enabled printers I find during engagements is astounding.

True, since nobody cares about them too much (yet). It's a printer. It prints. That's it. right ? 😉

> Sirius:
>
> Excellent write-up

You're welcome and thanks a lot!

2

pry0cc Leader & Offsec Engineer & Forum Daddy — Sep '17

Mate this is a sweet and very very complete write-up. Awesome job on this 😉 I would also love to know if this has worked for anybody else? Printer exploitation is a super difficult one to learn, and this is why I am so glad you've made it a LOT easier for people with this article.

Looking forward to your next bounty!

3

ricksanchez Leader — Sep '17

I may be able to share a detailed exploit in the near future. Depends on what I can "publicly disclose" and what not 😛
On top of my private investigations of course

2

pry0cc Leader & Offsec Engineer & Forum Daddy — Sep '17

Publicly disclose it all brother 😛

Remember we have a VIP section…

2 Replies

1

ricksanchez Leader Sep '17

Ah true! I'll come back to the VIP section in due time then 😬

3

2 MONTHS LATER

REal0day pry0cc Nov '17

What items should be placed in the VIP section and what items shouldn't?

11 DAYS LATER

DrewV Dec '17

Awsum #writeup. 😉 It's so useful! i enjoyed reading this.

1

ricksanchez Leader Dec '17

Thank you! Glad you enjoyed reading it 🙂

Raw-x Dec '17

Thanks for yor writting
now we can use eternalblue for port 445
Thank you NSA)))

CLOSED DEC 5, '17

16 DAYS LATER

OPENED DEC 22, '17

CLOSED DEC 22, '17

Reply

Suggested Topics

Topic	Replies	Activity
Windows 7 after the Supportend **Exploit Development** windows	**13**	4d
[KEYGEN] I hate rectangles **Challenges** keygen	**0**	Feb 2

Topic	Replies	Activity
New Things to 0x00sec! (Patreon & Stickers) **0x00sec Announcem…** patreon, stickers	**16**	7d
Psyche-reconnaissance journey - How well do you know yourself? **Philosophy**	**0**	Dec '18
How fun accidentally became security risk **Pentesting** hacking	**19**	15d

Want to read more? Browse other topics in Exploit Developm… **or view latest topics.**