Prism: Real-Time Privacy Protection Against Temporal Network Traffic Analyzers

Wenhao Li, Xiao-Yu Zhang, Huaifeng Bao, Binbin Yang, Zhaoxuan Li

Abril 2023

Contexto

- Análise de padrões em pacotes criptografados que prejudicam a segurança da rede

- A maior parte dos analisadores de tráfego aprendem via características temporais. ex: Sequências de tamanho de pacote e intervalos de chegada

- Propostas atuais empregam exemplos adversariais para ofuscar o tráfego.

Objetivo da pesquisa

- Propor um esquema de defesa que:
  - Engane analisadores temporais:
    - Targeted Defense: Confundir focando numa classe específica
    - Untargeted Defense: Defesa geral para todas as classes
  - Atue em tempo real
  - Seja assimétrico (Todo tráfego é manipulado em um único nó)

Problema de pesquisa

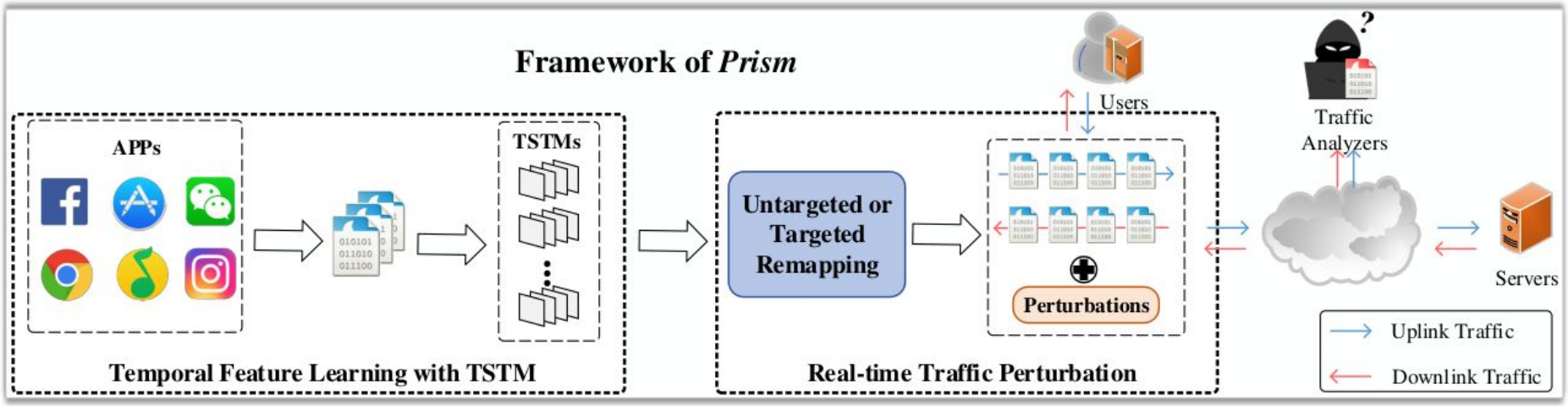
- Projetar as perturbações no fluxo de dados em tempo real

- Solução não deve sobrecarregar o fluxo de pacotes na rede.

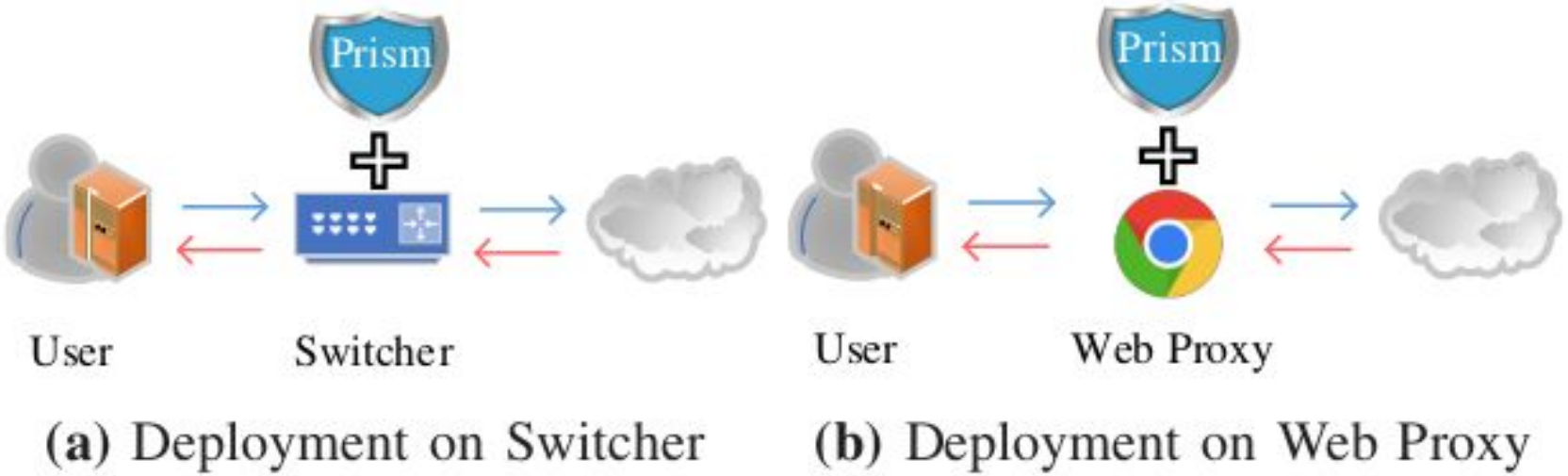
Proposta - PRISM

- Aprendizagem das características temporais
- Modelo de transição de estados
- Perturbação do tráfego

Proposta - PRISM



Proposta - PRISM



Temporal Feature Learning

- As características temporais são padronizadas com o algoritmo PLD (Power-Law Division Algorithm), que busca as características mais frequentes para cada classe de tráfego
  - Seleciona-se uma categoria de feature (ex: tamanho do pacote).
  - São coletados todos os pacotes com essa feature.
  - Para as outras features nos pacotes(da categoria) , calcula-se a frequência.
  - Ordena-se a lista de frequências.
  - São usadas as primeiras K características mais frequentes.

Time-Stacked State Transition Modeling (TSTM)

- Tendo as características mais frequentes, o objetivo é obter a assinatura (*fingerprint*) de cada classe.
- São usadas matrizes de time-steps diferentes para caracterizar as classes.
- É mantida uma matriz temporal para cada classe.
- Cada elemento da matriz é a probabilidade de transição de pacotes entre dois time-steps diferentes.
- As transições dos pacotes são utilizadas para identificar os fingerprints.

Partição dos fluxos em Packet Blocks

- Usa as ACK’s do protocolo TCP/IP como indicadores.
- Divide um fluxo de tráfego em diversos blocos, denominados Packet Blocks.
- As perturbações são geradas nos blocos, o Prism somente manipula o tráfego de um bloco por vez. De forma dinâmica e consecutiva.

Defesa global (Untargeted Defense)

- Para cada packet block, o objetivo é fazer com que um adversário o classifique com a classe diferente da correta (y).

- São usadas as transições minimizadas da matriz de probabilidades em cada time step para gerar os exemplos adversariais

- É gerada a sequência de características perturbadas com a probabilidade mínima de transição.

Defesa pontual (Targeted Defense)

- O objetivo é fazer com que uma classe específica não seja classificada corretamente.

- Para isso, ele aproxima a distribuição de características da classe alvo. Essa distribuição é usada para gerar os exemplos adversariais.

Experimentos

Base de dados

- ISCX2016: Tráfego de 15 aplicações (Youtube, Netflix, Facebook, etc). 207000 pacotes, tráfego totalmente criptografado

- USTC2016: Contém 10 tipos de tráfego de malware de sites de conexões reais, mas também tráfego de aplicações benignas (Skype, Facebook, etc).

Analisadores atacantes testados

TABLE I
TEMPORAL AND NON-TEMPORAL FEATURES USED BY THE ANALYZERS.

	Analyzers	Features
Temporal	FS-Net [13]	Sequences of message type and packet size.
	DeepCorr [1]	Inter-packet delays, intervals and packet sizes.
	WF-LSTM [51]	Directions of packet sequences.
	MB-Tree [16]	Sequences of the head and tailed packets size.
Non-Temporal	RBRN [10]	Byte Sequences of traffic flows.
	FC-Net [12]	Gray-scale images of raw-byte data flows.
	RF [53]	Statistical features such as max/min sizes.
	FlowPrint [22]	Statistical, temporal features and certificate.

			Without Perturbation				With Perturbation (Untargeted Defense)			
			Acc (%)	**Pre (%)**	**Re (%)**	**F1 (%)**	**Acc (%)**	**Pre (%)**	**Re (%)**	**F1 (%)**
Temporal (VI-A,VI-B)	ISCX2016	FS-Net [13]	86.77	88.59	87.47	88.03	3.92 (82.85↓)	3.07 (85.52↓)	2.93 (84.54↓)	3.00 (85.03↓)
		DeepCorr [1]	83.05	84.11	83.89	84.00	1.53 (81.52↓)	1.49 (82.62↓)	1.63 (82.26↓)	1.56 (82.44↓)
		MBTree [16]	90.67	88.64	88.38	88.51	0.78 (89.89↓)	1.21 (87.43↓)	0.89 (87.49↓)	1.03 (87.48↓)
		WF-LSTM [51]	62.89	61.21	62.86	62.02	5.32 (57.57↓)	3.05 (58.16↓)	3.94 (58.92↓)	3.44 (58.58↓)
	USTC2016	FS-Net [13]	96.03	97.32	96.88	97.10	2.48 (93.55↓)	2.03 (95.29↓)	2.15 (94.73↓)	2.09 (95.01↓)
		DeepCorr [1]	86.73	85.30	86.59	85.94	1.93 (84.80↓)	1.29 (84.01↓)	1.80 (84.79↓)	1.50 (84.44↓)
		MBTree [16]	93.87	91.07	91.65	91.36	1.05 (92.82↓)	0.94 (90.13↓)	1.10 (90.55↓)	1.01 (90.35↓)
		WF-LSTM [51]	70.82	68.59	69.01	68.80	4.37 (66.45↓)	4.08 (64.51↓)	3.91 (65.10↓)	3.99 (64.81↓)
Non-Temporal (VI-C)	ISCX2016	RBRN [10]	94.32	94.32	94.21	94.27	14.82 (79.50↓)	13.70 (80.62↓)	14.03 (80.18↓)	13.86 (80.41↓)
		FC-Net [12]	83.77	85.74	84.58	85.16	9.01 (74.76↓)	11.96 (73.78↓)	10.59 (73.99↓)	11.23 (73.93↓)
		RF [53]	76.82	77.29	76.94	77.11	6.92 (69.90↓)	5.70 (71.59↓)	5.42 (71.52↓)	5.56 (71.55↓)
		FlowPrint [22]	93.42	94.89	93.42	94.15	34.71 (58.71↓)	28.13 (66.76↓)	32.67 (60.75↓)	30.23 (63.92↓)
	USTC2016	RBRN [10]	93.30	94.19	94.86	94.52	11.92 (81.38↓)	10.53 (83.66↓)	10.47 (84.39↓)	10.50 (84.02↓)
		FC-Net [12]	84.55	85.13	85.30	85.21	7.68 (76.87↓)	7.24 (77.89↓)	6.99 (78.31↓)	7.11 (78.10↓)
		RF [53]	81.95	79.76	80.07	79.91	5.75 (76.20↓)	6.24 (73.52↓)	6.19 (73.88↓)	6.21 (73.70↓)
		FlowPrint [22]	95.42	94.69	94.72	94.70	27.73 (67.69↓)	25.08 (69.61↓)	26.91 (67.81↓)	25.96 (68.74↓)

Analisadores defensivos comparados

- BLANKET: Gera exemplos adversariais. Usa GAN. Analisa tamanho de pacotes,tempos e direções.

- WTF-PAD: Ofuscador de tráfego que adiciona pacotes ingênuos (*dummy packets*) para enganar analisadores

- Manipulator: Cria tráfego pela manipulação de intervalos, camadas de protocolos e tamanhos de pacotes. Não pode ser usado em tempo real.

		FS-Net [13]		DeepCorr [1]		MBTree [16]		WF-LSTM [51]	
		$R_{\mathrm{UD}}(\%)$	$R_{\mathrm{TD}}(\%)$	$R_{\mathrm{UD}}(\%)$	$R_{\mathrm{TD}}(\%)$	$R_{\mathrm{UD}}(\%)$	$R_{\mathrm{TD}}(\%)$	$R_{\mathrm{UD}}(\%)$	$R_{\mathrm{TD}}(\%)$
ISCX2016	BLANKET [3]	75.26	35.02	64.83	27.83	89.72	71.70	87.39	48.50
	WTF-PAD [37]	37.43	-	32.05	-	20.94	-	67.39	-
	Manipulator [47]	74.60	61.94	64.06	58.14	86.42	39.87	90.70	24.92
	Prism	**97.83**	**84.34**	**99.06**	**79.77**	**99.59**	**91.69**	**96.65**	**57.91**
USTC2016	BLANKET [3]	82.03	40.81	65.71	36.47	94.36	72.49	85.73	45.76
	WTF-PAD [37]	38.93	-	38.90	-	17.99	-	65.87	-
	Manipulator [47]	85.48	79.92	71.69	45.81	92.06	51.01	84.79	33.00
	Prism	**98.18**	**88.17**	**99.27**	**84.25**	**99.62**	**92.41**	**97.03**	**63.89**

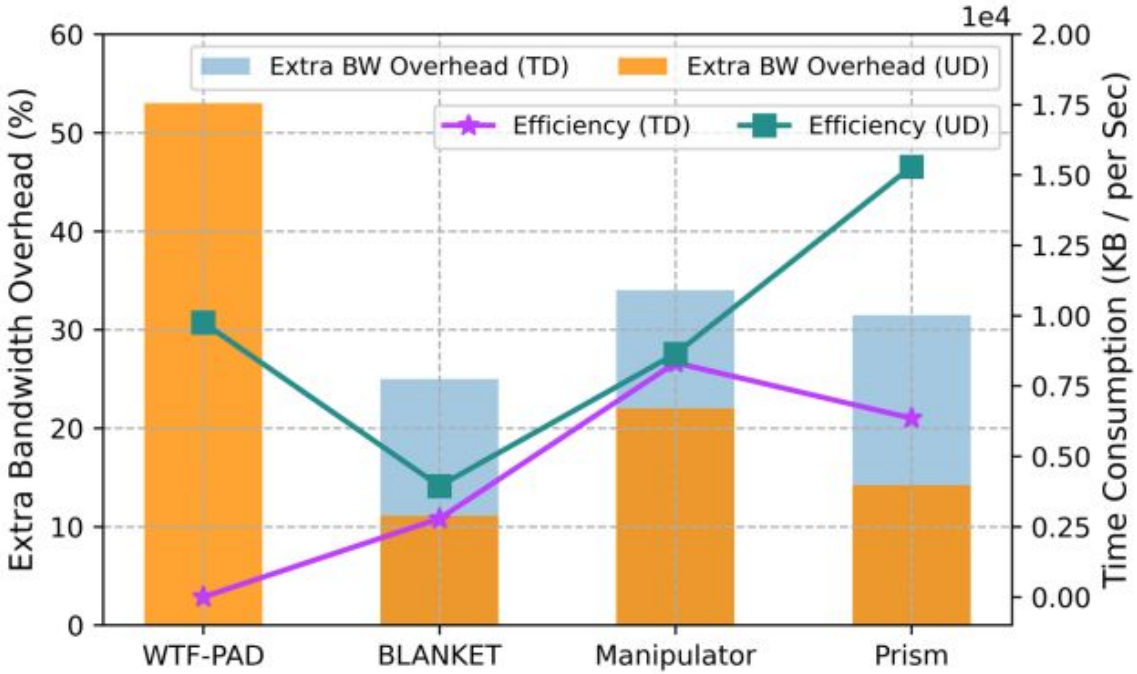


Fig. 8. Extra Bandwidth Overhead (BW Overhead) and Time Consumption when evaluating targeted defenses (TD) and untargeted defenses (UD) on FS-Net.

Sobre a implementação

- Desenvolvido em Python

- https://github.com/SecTeamPolaris/Prism

Duvidas?