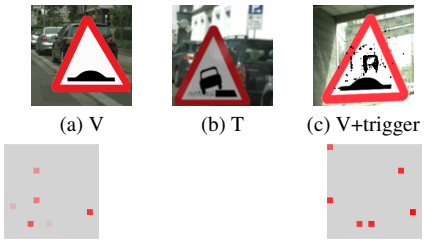
Appendix

A. An Example for the Necessity of Symmetry

Figure 7 presents an example for one-sided masks from a clean model #18 in TrojAI round 3. Figure 7 (a) and (b) present the victim and target classes and (c) a natural trigger (i.e., a trigger naturally exists and flips victim samples to the target label) generated by ABS, which resembles the central symbol in the target class. Figure (d) shows the one-sided mask from V to T, meaning that copying/mixing the feature maps as indicated by the mask from T samples to V samples can flip the classification results to T. Figure (e) shows the one-sided mask from V to V+trigger. Note that V+trigger samples are classified to T. Although in both cases V samples are flipped to T, the two one-sided masks have only one entry in common, suggesting that the ways they induce the classification results are different. In contrast, the symmetric masks share a lot of commonality.



(a) V (b) T (c) V+trigger

(d) 1 sided mask from V to T (e) 1 sided mask from V to V+trigger

Figure 7. One sided masks for a clean model # 18 in round 3 with victim class V =#8 and target class T =#3

B. Detecting Hidden-trigger Attack

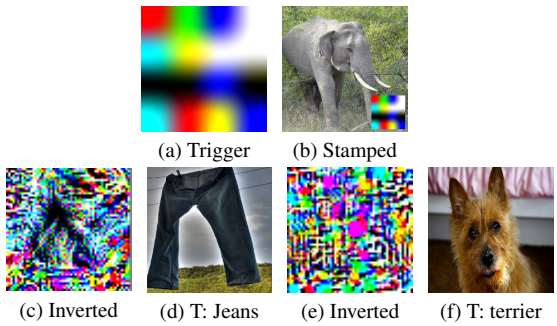
Table 5 shows the results on hidden-trigger attack. We use the optimal trigger size bound for ABS and 4,000 ($\approx$63$\times$63) for ABS+Ex-Ray. Observe that our technique can achieve 85% accuracy, surpassing ABS by 17%. Figure 8 (a) and (b) show an injected trigger and an example image stamped with the trigger, whose target label is terrier dog as shown in (f). Figure 8 (e) shows the generated trigger by ABS for the label terrier dog of the trojaned model. The trigger does not possess any features of the target label. It can hence be identified by Ex-Ray as a true positive. In contrast, Figure 8 (c) and (d) show a trigger by ABS for a benign model and its target label jeans. Observe that the central part of the trigger resembles a pair of jeans. Ex-Ray hence can flag the model as a true negative.

C. Detecting WaNet and Input-aware Dynamic Attacks

In this section, we evaluate Ex-Ray on WaNet [45] and input-aware dynamic [46] attacks. We utilize the CIFAR10 dataset and evaluate on 20 benign models and 40 trojaned models (20 trojaned models for each attack). We set the bound of the trigger size to be 12.5% of the input. The results are shown in Table 6. Observe that Ex-Ray achieves 82.5% detection accuracy on WaNet and 87.5% detection accuracy on Input-aware.

Table 5. Ex-Ray on hidden-trigger attack

	ABS					ABS+Ex-Ray				
	TP	FP	FN	TN	Acc/ROC	TP	FP	FN	TN	Acc/ROC
ImageNet	12	6	5	11	0.68	15	3	2	14	0.85



(a) Trigger (b) Stamped

(c) Inverted (d) T: Jeans (e) Inverted (f) T: terrier

Figure 8. A case for hidden-trigger attack

Table 6. ABS + Ex-Ray on WaNet and input-aware dynamic attacks

	TP	FP	FN	TN	Acc/ROC
Wanet	17	2	5	17	0.825
Input-aware	17	2	3	18	0.875

D. Comparison with other SOTA Defenses on Complex Backdoors

We compare Ex-Ray with four other state-of-the-art (SOTA) defenses, namely, Meta-Neural Analysis [73], DeepInspect [11], NeuronInspect [24], and TABOR [21], on reflection and composite attacks.

We use CIFAR10 and conduct experiments on 20 benign models, 20 trojaned models by composite attack, and 20 trojaned models by reflection attack. Meta-Neural Analysis outputs a binary result denoting whether a model is trojaned or not. The other three methods output a median absolute deviation (MAD) score for each model, which is used to distinguish benign and trojaned models. For DeepInspect, NeuronInspect, and TABOR, we search for the best possible bound of MAD scores for separating benign and trojaned models. Table 7 shows the results. Rows 2-9 show the results of Meta-Neural Analysis, DeepInspect, NeuronInspect, and TABOR on composite and reflection attacks. Rows 10-11 show the result of ABS + Ex-Ray. Observe that ABS+Ex-Ray outperforms the SOTA methods, hav-

Table 7. Comparison between EX-RAY and other defenses on composite and reflection attacks

		TP	FP	FN	TN	Acc/ROC
Meta-Neural Analysis	Composite	15	6	5	14	0.73
	Reflection	11	8	9	12	0.58
DeepInspect	Composite	20	19	0	1	0.53
	Reflection	20	20	0	0	0.5
NeuronInspect	Composite	4	0	16	20	0.6
	Reflection	2	0	18	20	0.55
TABOR	Composite	3	0	17	20	0.58
	Reflection	2	0	18	20	0.55
ABS+EX-RAY	Composite	17	3	3	17	0.85
	Reflection	18	4	2	16	0.85

ing at least 12% better accuracy on composite backdoors and 27% better on reflection backdoors.

During the TrojAI competition, performers tried many different SOTA methods [11, 16, 25, 29, 55, 57, 58, 60, 73] (including DeepInpect, Meta-Neural Analysis and K-Arm). Except for K-Arm [55], all other methods perform worse than ABS + EX-RAY in rounds 2 to 4. K-Arm performs better than ABS + EX-RAY in round 3 but worse than ABS + EX-RAY in rounds 2 and 4.

E. Using EX-RAY with Different Upstream Scanners on Complex Backdoors

We apply EX-RAY to different upstream scanners, including Neural Cleanse (NC) [66] and K-Arm [55], on detecting composite and reflection backdoors on CIFAR10. During the detection, we first use NC/K-Arm to invert triggers and then apply EX-RAY to determine whether a model is trojaned or not. The results are shown in Table 8. The first column denotes the detection methods. The second column shows the different attacks. Columns 3-7 show the detection results by vanilla NC and vanilla K-Arm. Columns 8-12 show the detection results by NC+EX-RAY and K-Arm+EX-RAY. Observe that EX-RAY can improve NC's detection accuracy from 50-60% to 68-75%, and K-Arm's from 55-58% to 73-75%. We also evaluate the combinations of EX-RAY and different upstream scanners on the TrojAI datasets. We use NC and Bottom-up-Top-down method [1] (used in the TrojAI competitions) as the upstream scanners. The results show that EX-RAY can consistently improve NC by 25%, and Bottom-up-Top-down by 2-15%. Please see more details in Appendix J.

F. Description of TrojAI and ImageNet Datasets

We use TrojAI rounds 2-4 training and test datasets [3]. EX-RAY *does not* require training and hence we use both training and test sets as regular datasets in our experiments.

Table 8. EX-RAY with different upstream scanners on composite and reflection attacks

		Vanilla					+EX-RAY				
		TP	FP	FN	TN	Acc/ROC	TP	FP	FN	TN	Acc/ROC
NC	Composite	7	3	13	17	0.60	14	4	6	16	0.75
	Reflection	5	5	15	15	0.50	10	3	10	17	0.68
K-Arm	Composite	3	1	17	19	0.55	16	6	4	14	0.75
	Reflection	18	15	2	5	0.58	16	7	4	13	0.73

TrojAI round 2 training set has 552 clean models and 552 trojaned models with 22 structures. Each TrojAI model has its own unique dataset. The data are mostly synthetic traffic signs with some street view background. A traffic sign is a polygon of solid color with some symbol in the center. The models are classifiers for the different kinds of signs. TrojAI has two types of backdoors: polygons (i.e., static patch triggers) and Instagram filters (i.e., dynamic and pervasive triggers). Round 2 test set has 72 clean and 72 trojaned models. Most performers had difficulties for round 2 due to the prevalence of natural triggers, which are small triggers that naturally exist and can flip classification results among benign labels. IARPA hence introduced adversarial training [43, 70] in round 3 to enlarge the distance between classes and suppress natural triggers. Round 3 training set has 504 clean and 504 trojaned models and the test set has 144 clean and 144 trojaned models. In round 4, triggers may be position dependent, meaning that they only cause misclassification when stamped at a specific position inside the foreground object. A model may have multiple backdoors. The number of clean images provided is reduced from 10-20 (in rounds 2 and 3) to 2-5. Its training set has 504 clean and 504 trojaned models and the test set has 144 clean and 144 trojaned models. Training sets were evaluated on our local server whereas test set evaluation was done remotely by IARPA on their server.

We also use a number of models on ImageNet. They have the VGG, ResNet and DenseNet structures. We use 7 trojaned models from [39] and 17 pre-trained clean models from torchvision zoo [2].

G. Parameter Settings

EX-RAY has three hyper-parameters, α to control the weight changes of cross-entropy loss in function (5) (in the design section) , β to control the similarity comparison between masks in condition (6) in Section 3.2, and γ the accuracy threshold in cross-validation checks of masks in Section 3.2. We use 0.1, 0.8, and 0.8, respectively, by default. In our experiments, we use ABS and NC as the upstream scanners. The numbers of optimization epochs are 60 for ABS and 1000 for NC. The other settings are default unless stated otherwise. The experiments are all done on an identical machine with a single 11GB memory NVIDIA RTX

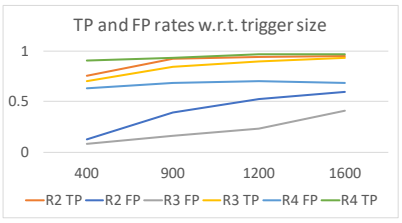


Figure 9. Rounds 2-4 true positive rates (TPs) and false positive rates (FPs) versus trigger size (in pixels) by ABS

2080Ti GPU (with the lab server configuration), except for the TrojAI test sets that are run on IARPA server with a single 32GB memory NVIDIA V100 GPU.

H. Experiments on TrojAI and ImageNet Models

In the first experiment, we evaluate EX-RAY on TrojAI rounds 2-4 training sets and the ImageNet models. We do not include TrojAI test sets in this experiment as the test sets were evaluated on an IARPA server and the results are reflected on the leaderboard. Here we use ABS as the upstream scanner as it is much faster than NC.

A critical setup for scanners that produce triggers, such as ABS and NC, is the maximum trigger size. A large value enables detecting injected backdoors with large triggers, while producing a lot of natural triggers and hence false positives. Figure 9 studies how the true positives (TPs) and false positives (FPs) change with different trigger bounds in the vanilla ABS, on the TrojAI rounds 2-4 training sets. Observe that both grow with the trigger size. Observe that there is a lower FP rate in round 3 (compared to round 2), illustrating the effect of adversarial training, although the number is still large when the trigger size is large. Round 4 has the highest FP rate because the number of clean images available is decreased and it is hence very easy for scanners to find (bogus) triggers that can induce misclassification on all the available images.

Based on the study, we use the trigger size bound 900 pixels for round 2, 1600 pixels for round 3, and 1200 pixels for round 4 for our experiment such that the upstream scanner does not miss many true positives to begin with and we can stress test EX-RAY.

Baselines. In the experiment, we compare EX-RAY against 8 baselines. The first baseline is using L2 distance of inner activation between $V+t$ and T. Such a distance for a natural trigger is supposed to be smaller than that of an injected trigger . We use unsupervised learning to report the best separation. In the second baseline, we use half of the models to train a random forest classifier based on the inner activations and logits values to distinguish natural and injected triggers and test it on the other half. Specifically, the classifier takes the L2 distance between $V+t$ and T, L2 distance between $T+t$ and T, L2 distance between V and T, logits of V, logits of $V+t$ and logits of T. The third baseline uses *integrated gradients (IG)* [59], an attribution technique, to find important neurons for $V+t$ and for T and then apply the aforementioned L2 distance comparison on the 10% most important neurons . Originally, integrated gradients were used in model explanation to identify important pixels. We adapt it to work on inner layers to identify important neurons. The next three baselines are similar to the third except having different methods to identify important neurons. Specifically, the fourth baseline uses *Deeplift* [56], the fifth uses *Occlusion* [5] and the sixth uses *Network Dissection (NE)* [8]. For baselines 4-7, we use unsupervised learning to find the best separation (of natural and injected backdoors). We will release the settings together with our system upon publication. EX-RAY is symmetric. To study the necessity of symmetry, the seventh and eighth baselines are one-sided versions of EX-RAY, that is, requiring satisfying either constraint (1) or (2) in the design section.

The results are shown in Table 9. The first column shows the methods. The first method is the vanilla ABS. Columns 2-4 show the results for TrojAI round 2 models with polygon backdoors. Column 2 shows the number of true positives (TPs). Note that there are 276 trojaned models with polygon backdoors. As such the vanilla ABS having 254 TPs means it has 22 false negatives. Column 3 shows the number of false positives (FPs) out of the 552 clean models. Column 4 shows the overall detection accuracy (on the total 552+276=828 models). Columns 5-7 show the results for round 2 models with Instagram filter backdoors. ABS uses a different method for filter backdoors. Instead of reverse engineering a pixel patch, it reverse engineers a one-layer kernel denoting general filter transformation [39]. Hence, we separate the evaluation of EX-RAY on the two kinds of backdoors. Note that the accuracy is computed considering the same 552 clean models. The overall results (for all kinds of backdoors) on the leaderboard are presented in the main text. Columns 8-13 show the results for round 3 and columns 14-19 for round 4. Columns 14-16 show the results for ImageNet patch attack.

The results show that the vanilla ABS has a lot of FPs (in order not to lose TPs) and EX-RAY can substantially reduce the FPs by 78-100% with the cost of increased FNs (i.e., losing TPs) by 0-30%. The overall detection accuracy improvement (from vanilla ABS) is 17-41% across the datasets. Also observe that EX-RAY consistently outperforms all the baselines, especially the non-EX-RAY ones. Attribution techniques can remove a lot of natural triggers indicated by the decrease of FPs. However, they preclude many injected triggers (TPs) as well, leading to inferior performance. The missing entries for NE are because it requires an input region to decide important neurons, rendering it inapplicable to filters that are pervasive. Symmetric EX-RAY outperforms the one-sided versions, suggesting

Table 9. Effectiveness of EX-RAY; (T:276,C:552) means that there are 276 trojaned models and 552 clean models

	TrojAI R2						TrojAI R3						TrojAI R4						ImageNet		
	Polygon trigger (T:276,C:552)			Filter trigger (T:276,C:552)			Polygon trigger (T:252,C:504)			Filter trigger (T:252,C:504)			Polygon trigger (T:143,C:504)			Filter trigger (T:361,C:504)			Patch Trigger (T:7, C:17)		
	TP	FP	Acc	TP	FP	Acc	TP	FP	Acc	TP	FP	Acc	TP	FP	Acc	TP	FP	Acc	TP	FP	Acc
Vanilla ABS	254	218	0.710	260	293	0.626	235	208	0.702	213	334	0.528	137	355	0.442	331	376	0.531	7	7	0.708
Inner L2	188	93	0.782	153	123	0.703	210	111	0.798	133	123	0.680	73	137	0.680	208	217	0.572	7	0	1
Inner RF	192	76	0.807	196	101	0.781	159	46	0.816	153	110	0.724	133	265	0.575	330	353	0.556	7	0	1
IG	172	29	0.840	192	66	0.818	162	58	0.804	52	41	0.681	84	53	0.827	210	87	0.725	5	0	0.917
Deeplift	152	11	0.837	189	21	0.869	162	59	0.803	78	67	0.681	84	54	0.825	203	54	0.755	6	0	0.958
Occulation	173	24	0.847	207	47	0.860	164	58	0.807	78	66	0.683	85	52	0.830	251	107	0.749	7	3	0.875
NE	180	58	0.814	-	-	-	187	72	0.819	-	-	-	59	72	0.759	-	-	-	7	4	0.833
1-sided(V to T)	157	19	0.833	195	33	0.862	202	62	0.852	153	51	0.802	107	82	0.818	236	50	0.798	7	0	1
1-sided(T to V)	134	4	0.824	158	18	0.835	187	50	0.848	134	27	0.808	102	56	0.850	179	9	0.779	1	1	0.958
EX-RAY	198	19	0.883	204	32	0.874	200	46	0.870	149	39	0.812	105	53	0.859	242	46	0.809	7	0	1

the need of symmetry.

Runtime. EX-RAY's time complexity is $\Omega(n)$ with n the number of triggers the upstream technique generates. Our upstream ABS uses neuron stimulation analysis to select three most likely target labels for each (victim) label for trigger inversion. It also filters out large sized triggers. In TrojAI, the number of classes per model varies from 15 to 45. On average, EX-RAY takes 12s to process a trigger, 95s to process a model. ABS takes 337s to process a model, producing 8.5 triggers on average.

I. Effects of Hyperparameters

We study EX-RAY performance with various hyperparameter settings, including the different layer to which EX-RAY is applied, different trigger size settings (in the upstream scanner) and different SSIM score bounds (in filter backdoor scanning to ensure the generated kernel does not over-transform an input), and the α, β, and γ settings of EX-RAY. Table 11 shows the results for layer selection. The row "Middle" means that we apply EX-RAY at the layer in the middle of a model. The rows "Last" and"2nd last" show the results at the last and the second-last convolutional layers, respectively. Observe that layer selection may affect performance to some extent and the second to the last layer has the best performance. Table 10 shows the results with and without the additional validation checks. Observe that removing the additional validation check results in 0.7% to 3% decrease in detection accuracy. Tables 12 shows that a large trigger size degrades EX-RAY's performance but EX-RAY is stable in 900 to 1200. Table 13 shows that the SSIM score bound has small effect on performance in 0.7-0.9. Note that an SSIM score smaller than 0.7 means the transformed image is quite different (in human eyes). Figures 10, 11, and 12 show the performance variations with α, β, and γ, respectively. The experiments are on the mixture of trojaned models with polygon triggers and the

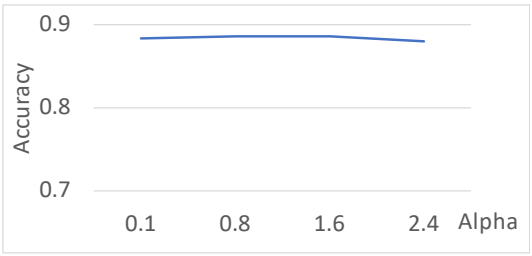


Figure 10. Accuracy changes with α on TrojAI R2

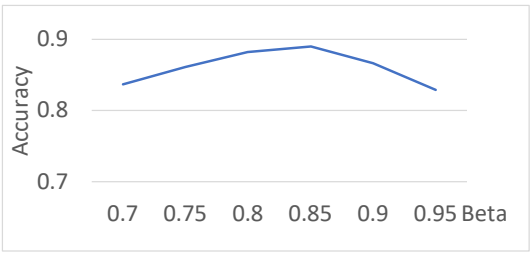


Figure 11. Accuracy changes with β on TrojAI R2

clean models from TrojAI round 2. For β and γ, we sample from 0.7 to 0.95 and for α we sample from 0.1 to 2.4. Observe that changing α and γ does not have much impact on the overall accuracy. When we change β from 0.7 to 0.95, the overall accuracy is still consistently higher than 0.83. These results show the stability of EX-RAY.

J. Using EX-RAY with Different Upstream Scanners on TrojAI dataset

In this experiment, we use EX-RAY with different upstream scanners, including Neural Cleanse (NC) [66] and the Bottom-up-Top-down method by the SRI team in the TrojAI competition [1]. The latter has two sub-components, trigger generation and a classifier that makes use of features collected from the trigger generation process. We created two scanners out of their solution. In the first one, we ap-

Table 10. EX-RAY w. and w.o. additional check; (T:276,C:552) means that there are 276 trojaned models and 552 clean models

	TrojAI R2						TrojAI R3						TrojAI R4					
	Polygon Trigger (T:276,C:552)			Filter Trigger (T:276,C:552)			Polygon Trigger (T:252,C:504)			Filter Trigger (T:252,C:504)			Polygon trigger (T:143,C:504)			Filter trigger (T:361,C:504)		
	TP	FP	Acc	TP	FP	Acc	TP	FP	Acc	TP	FP	Acc	TP	FP	Acc	TP	FP	Acc
W. additional check	198	19	0.883	204	32	0.874	200	46	0.870	149	39	0.812	105	53	0.859	242	46	0.809
W.o. additional check	206	33	0.876	216	71	0.844	207	71	0.843	158	62	0.793	110	77	0.829	275	93	0.793

Table 11. EX-RAY with different layer; (T:276,C:552) means that there are 276 trojaned models and 552 clean models

	TrojAI R2						TrojAI R3						TrojAI R4					
	Polygon Trigger (T:276,C:552)			Filter Trigger (T:276,C:552)			Polygon Trigger (T:252,C:504)			Filter Trigger (T:252,C:504)			Polygon trigger (T:143,C:504)			Filter trigger (T:361,C:504)		
	TP	FP	Acc	TP	FP	Acc	TP	FP	Acc	TP	FP	Acc	TP	FP	Acc	TP	FP	Acc
Middle	215	54	0.861	220	97	0.815	212	55	0.874	153	58	0.792	102	75	0.821	257	77	0.791
Second Last	198	19	0.883	204	32	0.874	200	46	0.870	149	39	0.812	105	53	0.859	242	46	0.809
Last	141	6	0.83	171	16	0.854	193	55	0.849	141	27	0.817	84	37	0.852	196	37	0.766

Figure 12. Accuracy changes with γ on TrojAI R2

Table 12. EX-RAY with different trigger sizes; (T:276, C:552) means there are 276 trojaned models and 552 clean models

	TrojAI R2 (T:276,C:552)			TrojAI R3 (T:252,C:504)			TrojAI R4 (T:143,C:504)		
	TP	FP	Acc	TP	FP	Acc	TP	FP	Acc
900	198	19	0.883	157	19	0.849	95	58	0.836
1200	203	30	0.876	175	39	0.847	105	53	0.859
1600	210	46	0.864	200	46	0.870	108	77	0.827

Table 13. EX-RAY with different SSIM scores; (T:276, C:552) means there are 276 trojaned models and 552 clean models

SSIM Score	TrojAI R2 (T:276,C:552)			TrojAI R3 (T:252,C:504)			TrojAI R4 (T:361,C:504)		
	TP	FP	Acc	TP	FP	Acc	TP	FP	Acc
0.9	145	4	0.837	115	9	0.807	234	47	0.799
0.8	160	13	0.844	149	39	0.812	242	46	0.809
0.7	204	32	0.874	178	90	0.783	175	13	0.770

ply EX-RAY on top of their final classification results (i.e., using EX-RAY as a refinement). We call it SRI-CLS. In the second one, we apply EX-RAY right after their trigger generation. We have to replace their classifier with the simpler unsupervised learning (i.e., finding the best separation) as adding EX-RAY changes the features and nullifies their original classifier. We call it SRI-RE. We use the round 2 clean models and models with polygon backdoors to conduct the study as NC does not handle Instagram filter triggers. For SRI-CLS, the training was on 800 randomly selected models and testing was on the remaining 146 trojaned models and 158 clean models. The other scanners do not require training. The results are shown in Table 14. The T and C columns stand for the number of trojaned and clean models used in testing, respectively. Observe that the vanilla NC identifies 180 TPs and 332 FPs with the accuracy of 44.7%. With EX-RAY, the FPs are reduced to 73 (81.1% reduction) and the TPs become 127 (29.4% degradation). The overall accuracy improves from 44.7% to 70.8%. The improvement for SRI-RE is from 53.6% to 68.5%. The improvement for SRI-CLS is relative less significant. That is because 0.882 accuracy is already very close to the best performance for this round. The results show that EX-RAY can consistently improve upstream scanner performance. Note that the value of EX-RAY lies in suppressing false warnings. It offers little help if the upstream scanner has substantial false negatives. In this case, users may want to tune the upstream scanner to have minimal false negatives and then rely on the downstream EX-RAY to prune the false positives like we did in the ABS+EX-RAY pipeline.

K. Mask and Differential Features

In this experiment, we aim to demonstrate that the masks computed by EX-RAY indeed capture the feature differences. Specifically, we want to show that 1) *the mask between $V + t$ and V covers the trigger features and does not overlap with the natural differences between V and T*

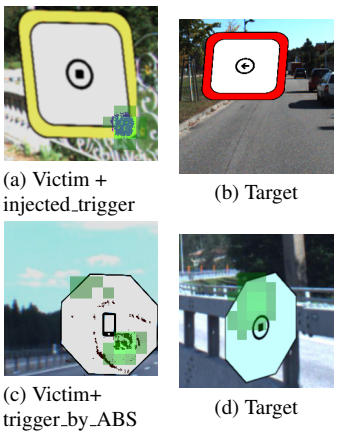
for a trojaned model and 2) *the mask between $V+t$ and V captures the natural differences between V and T for a clean model*. We use a model interpretation technique similar to [8] to project the large activation values of feature maps (i.e., neurons) in the mask between $V+t$ and V back to the input space and observe which input areas are highlighted. Figure 13 shows a sample result. Figures (a)-(b) correspond to a trojaned model #7 in TrojAI round 2. Figure (a) shows a victim sample with the trigger, which is a purple polygon. The area in light-green shows the input area corresponding to the activated neurons in the mask. Observe that the two align nicely, indicating the mask captures the trigger features. In contrast, figure (b) shows a target sample and also the input area corresponding to activated neurons in the mask, if any. Observe that those neurons do not have large activations. Figures (c)-(d) show the results for a clean model #123. Figure (c) shows that the (natural) trigger is to the right and below the central symbol of the victim sample. Observe that while there is a highlighted area in the $V+t$ sample (c) covering the trigger, there is also a highlighted area in the T sample (d) covering the target features, demonstrating that the mask between $V+t$ and V indeed captures T's features. Figure 14 plots the maximum activation values for all the neurons in the mask, with (a)-(c) for model #7 and (d)-(f) for model #123. For example, a data point in (a) shows the average maximum activation of a neuron in the mask for all $V+t$ samples (y axis), versus its average maximum activation for all V samples (x axis). From (a)-(c), we can observe that these neurons are substantially activated when t is present but never for clean V or T samples. In contrast, (d)-(f) show that the neurons in the mask are activated when either t is present or V/T samples are provided. We studied a few other models. Their results are similar and hence omitted.

L. Two Additional Adaptive Attacks

EX-RAY is not a stand-alone defense technique and supposed to be part of an end-to-end scanning pipeline. We devise another the second attack that forces the internal activations of victim class inputs embedding the trigger to resemble the activations of the target class inputs such that EX-RAY cannot distinguish the two. In particular, we train a Network in Network model on CIFAR10 with an 8×8 patch trigger. In order to force the activations of images stamped with the trigger to resemble those of target class

Table 14. EX-RAY with different upstream scanners

	Vanilla					+EX-RAY					
	TP	T	FP	C	Acc	TP	T	FP	C	Acc	Acc Inc
NC	180	252	332	552	0.483	127	252	73	552	0.732	0.249
SRI-RE	164	252	272	552	0.536	112	252	97	552	0.685	0.149
SRI-CLS	120	146	17	158	0.858	119	146	9	158	0.882	0.024

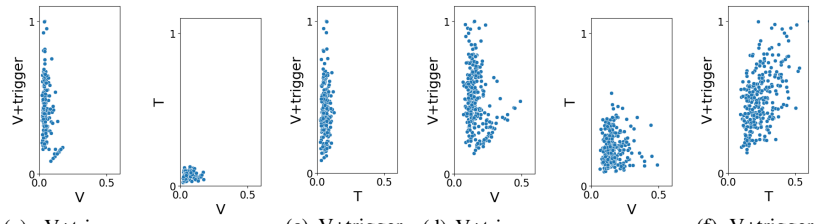


(a) Victim + injected_trigger (b) Target (c) Victim+ trigger_by_ABS (d) Target

Figure 13. Trojaned model #7 from TrojAI round 2 in (a)-(b) and clean model #123 in (c)-(d). In (a), the trigger is stamped at the right-bottom corner with the light-green area corresponding to the neurons in the mask by an interpretation technique; (b) shows the neurons in the mask do not have large activations at all for a target sample; (c) and (d) show that the neurons in the mask capture features in both the trigger and the target.

images, we design an adaptive loss to minimize the differences between the two. In particular, we measure the differences of the means and standard deviations of feature maps. During training, we add the adaptive loss to the normal cross-entropy loss. The effect of adaptive loss is controlled by a weight value, which essentially controls the strength of attack as well. Besides the adaptively trojaned model, we also train 20 clean models on CIFAR10 to see if ABS+EX-RAY can distinguish the trojaned and clean models. The results are shown in Table 15. The first row shows the adaptive loss weight. A larger weight value indicates stronger attack. The second row shows the trojaned model's accuracy on clean images, including both the overall accuracy and the victim label accuracy. The third row shows the attack success rate of the trojaned model. The fourth row shows the FP rate. The fifth row shows the TP rate. Observe while ABS+EX-RAY does not miss trojaned models, its FP rate grows with the strength of attack. When the weight value is 1000, the FP rate of ABS+EX-RAY becomes 0.65 while its TP rate remains 1, meaning effectiveness degradation. However at this setting, the model accuracy has degraded so much that such model is unlikely used in practice.

In the third adaptive attack, we first generate a trigger similar to a third class while having similar feature-level representations to the target class. We generate such triggers by optimizing two losses. The first is the cross entropy loss between the model output on images stamped with the trigger and the third class label (similar to adversarial noise for a third class). The second loss is the mean squared error loss between the inner activation of the images stamped with the trigger and the inner activation of the target class



(a) V+trigger v. V (b) T v. V (c) V+trigger v. T (d) V+trigger v. V (e) T v. V (f) V+trigger v. T

Figure 14. Average maximum activation values for neurons in the computed masks for a trojaned model #7 in (a)-(c), and for a clean model #123 in (d)-(f).

Table 15. The second adaptive attack

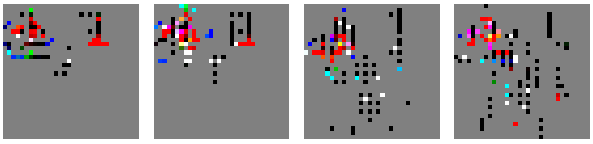
Weight of adaptive loss	1	10	100	200	400	600	800	1000	10000
Acc (model/label)	0.89/0.73	0.88/0.73	0.87/0.7	0.87/0.7	0.86/0.69	0.845/0.66	0.84/0.66	0.82/0.64	0.1
ASR	0.99	0.99	0.99	0.98	0.94	0.98	0.96	0.97	-
FP/ # of clean models	0	0.2	0.2	0.2	0.35	0.45	0.6	0.65	-
TP/ # of clean models	1	1	1	1	1	1	1	1	-

images (similar to adversarial feature-level attack). After generating the triggers, we use data poisoning to trojan the models. We do the experiment on CIFAR10. We choose label 0 as the target label and label 8 as the third label. We choose conv7 in NiN models as the feature layer and optimize neuron activations in this layer. We find that we need to enlarge the trigger size to have similar inner activations as the target label images. We generate triggers with 4 different sizes, 120, 140, 160, 200. The triggers are shown in Figure 15. We train 20 benign NiN models and 20 feature-level adaptive attack NiN models for each trigger size.

Table 16 shows the results of EX-RAY. Row 1 shows the different trigger sizes. Row 2 shows the mean squared activation differences. Observe that with the increase of trigger size, we can optimize the difference to a smaller value. A trigger with a small feature difference may be difficult to be detected. Rows 3 and 4 show the false positive and true positive rates. Observe that EX-RAY has 75% true positive rate when the trigger is 160 and 65% true positive rate when trigger size is 200. When the trigger size is 200, the trigger already covers a large part of the image. The attack becomes less meaningful.

Table 16. The third adaptive attack

Trigger size	120	140	160	200
Mean squared feature difference	0.153	0.116	0.034	0.009
FP/ # of clean models	0.1	0.1	0.1	0.1
TP/ # of clean models	1	0.8	0.75	0.65



(a) 120 (b) 140 (c) 160 (d) 200

Figure 15. Triggers in the third adaptive attack

M. Fixing Models with Injected and Natural Backdoors

In this experiment, we try to fix 5 benign models and 5 trojaned models on CIFAR10. Fixing a benign model means enlarging class distances to make it less vulnerable to (small) natural backdoors. The trojaned models are trojaned by label-specific data poisoning. Here we use unlearning [66] which stamps triggers generated by scanning methods on images of victim label to finetune the model and forces the model to unlearn the correlations between the triggers and the target label. The process is iterative, bounded by the level of model accuracy degradation. The level of repair achieved is measured by the trigger sizes of the fixed model. Larger triggers indicate the corresponding backdoors become more difficult to exploit. The trigger size increase rate suggests the difficulty level of repair.

Table 17 shows the average accuracy and average reverse engineered trigger size before and after fixing the models. All models have the same repair budget. We can see that natural triggers have a larger accuracy decrease. Natural trigger size only increases by 34.4 whereas injected trigger size increases by 78.

We show the trigger size for each label pair for an trojaned model in Figure 16 and for an benign model in Figure 17. Figure 16 (a) shows the trigger size between each

Table 17. Average trigger size change before and after unlearning

	Natural Trigger		Injected Trigger	
	Before	After	Before	After
Avg Acc	88.7%	85.9%	86.4%	85.4%
Avg Trigger Size	25.8	60.2	19	97

pair of labels. The columns denote the victim label and the rows denote the the target label. For example, the gray cell in Figure 16 (a) shows the trigger size to flip class 1 to class 0. Figure 16 (b) follows the same format and shows the result for a trojaned model after unlearning. In the trojaned model the injected trigger flips class 1 to class 0. Before unlearning, class 1 and class 0 have the smallest trigger size 21. Unlearning increases the trigger size between the two to 106, which is above the average trigger size between any pairs. Intuitively, one can consider the backdoor is fixed. In the benign model, the natural trigger flips class 3 to class 5. As shown in Figure 17, unlearning increases the trigger size from 24 to 59 and 59 is still one of the smallest trigger size among all label pairs for the fixed model. It demonstrates that natural backdoors are inevitable and difficult to fix. AI model users can use EX-RAY to find injected backdoors and speed up fixing process by prioritizing fixing injected backdoors first.

Note that model repair is not the focus of the paper and trigger size may not be a good metric to evaluate repair success for the more complex semantic backdoors. The experiment is to provide initial insights. A thorough model repair solution belongs to our future work.

(a) Before unlearning

	0	1	2	3	4	5	6	7	8	9
0	-	48	34	75	42	52	62	46	48	52
1	21	-	74	91	88	96	72	80	81	45
2	32	54	-	66	39	57	54	61	78	60
3	34	53	35	-	42	27	46	50	72	47
4	29	45	29	49	-	36	46	48	63	48
5	40	70	35	46	43	-	53	49	81	56
6	29	48	23	41	44	61	-	66	70	59
7	40	77	55	78	40	52	81	-	82	60
8	21	44	42	75	50	59	60	65	-	47
9	29	62	78	85	69	70	73	62	73	-

(b) Before unlearning

	0	1	2	3	4	5	6	7	8	9
0	-	84	56	103	77	96	82	96	56	78
1	106	-	132	162	150	140	113	134	124	70
2	92	111	-	88	79	79	62	99	109	106
3	105	92	66	-	86	60	58	82	124	86
4	96	92	55	81	-	72	53	77	99	95
5	119	100	64	70	91	-	58	101	136	90
6	107	97	86	88	99	93	-	113	113	97
7	94	101	92	124	87	87	81	-	126	100
8	50	72	68	104	98	106	81	101	-	79
9	104	87	129	119	117	115	110	108	123	-

Figure 16. Injected trigger distance matrix before and after unlearning

	0	1	2	3	4	5	6	7	8	9
0	-	43	44	47	38	42	67	68	37	48
1	62	-	89	88	79	78	70	85	76	47
2	53	58	-	37	35	42	51	71	72	62
3	62	66	40	-	40	24	48	59	72	56
4	61	57	38	48	-	31	52	52	85	64
5	69	66	43	33	46	-	51	61	73	57
6	66	55	32	35	44	38	-	80	87	62
7	74	77	67	61	38	39	74	-	92	68
8	29	44	55	61	48	51	62	65	-	46
9	79	57	84	72	67	67	83	76	72	-

	0	1	2	3	4	5	6	7	8	9
0	-	79	56	89	63	90	65	99	59	76
1	120	-	134	114	123	154	77	119	122	48
2	95	101	-	75	58	74	59	82	121	82
3	104	98	57	-	58	59	40	90	124	80
4	104	100	60	88	-	79	50	73	117	79
5	95	91	58	84	76	-	52	77	131	86
6	114	115	77	129	102	89	-	131	137	93
7	104	120	110	103	68	92	66	-	129	78
8	36	61	74	80	66	112	66	86	-	70
9	128	109	117	103	112	120	73	124	105	-

Figure 17. Natural trigger distance matrix before and after unlearning